Thèse n° 10 404

The Learnability of the Grammar of Jazz: Bayesian Inference of Hierarchical Structures in Harmony

Présentée le 11 décembre 2020

au Collège des humanités
Digital and Cognitive Musicology Lab
Programme doctoral en humanités digitales

pour l'obtention du grade de Docteur ès Sciences

par

Daniel HARASIM

Acceptée sur proposition du jury

Prof. S. Süsstrunk, présidente du jury
Prof. M. A. Rohrmeier, Prof. T. J. O'Donnell, directeurs de thèse
Prof. M. Steedman, rapporteur
Prof. D. Sears, rapporteur
Prof. W. Gerstner, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

For the first time, we now have practical methods for developing computational models of human cognition that are based on sound probabilistic principles and that can also capture something of the richness and complexity of everyday thinking, reasoning and learning.

Thomas L. Griffiths,
Charles Kemp, &
Joshua B. Tenenbaum

Dedicated to my father André Harasim and my love Urszula Bitner.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors Martin Rohrmeier and Timothy O'Donnell for their guidance, support, and all the feedback I received during my doctoral studies. Without Martin's constant encouragement to expand my horizon beyond the boarders of established academic disciplines, and his focus on never losing sight of the big picture, this work would not have been possible. Tim's motivation and persistence to concentrate on every detail of formal modeling and scientific writing is a model for me without which I would have achieved much less.

I appreciate the time and effort all members of my doctoral committee invested into reading and evaluating my thesis: Sabine Süsstrunk, Mark Steedman, David Sears, and Wulfram Gerstner. Thank you for your feedback and your helpful comments.

Furthermore, I would like to thank all members of the Digital and Cognitive Musicology Lab (DCML) for providing an incredibly friendly and inspiring environment in which I very much enjoyed carrying out the present research. In particular, many thanks are due to Christoph Finkensiep for his invaluable comments and suggestions on earlier drafts of this thesis, and to Fabian Moss, Petter Ericson, Robert Lieck, Andrew McLeod, and Steffen Herff for supporting me on the last meters.

I am equally glad to have met so many colleagues, mentors, and friends I learned from and discussed or worked with in the past years, Christine Ahrends, Colin Aitken, Rie Asano, Oren Boneh, Fernando Bravo, Sebastian Klaßmann, Thomas Noll, Jessica Pidoux, Tudor Popescu, Uri Rom, Stefan Schmidt, Oliver Schwab-Felisch, and Jonathan Walter (in alphabetical order).

My deepest appreciation goes to my grandparents for always encouraging me to be curious and learn about the world, and to my parents for their unconditional support and trust throughout my life. Finally, I am deeply grateful to my partner for being the anchor point in my life and backing me up ever since we met.

Lausanne, November 10, 2020 Daniel Harasim

Funding statement

The research presented in this thesis has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 760081 – PMSB. Furthermore, I thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology and the DCML.

Abstract

Musical grammar describes a set of principles that are used to understand and interpret the structure of a piece according to a musical style. The main topic of this study is grammar induction for harmony — the process of learning structural principles from the observation of chord sequences. The question how grammars are learnable by induction from sequential data is an instance of the more general question how abstract knowledge is inducible from the observation of data — a central question of cognitive science. Under the assumption that human learning approximately follows the principles of rational reasoning, Bayesian models of cognition can be used to simulate learning processes. This study investigates what prior knowledge makes it possible to learn musical grammar inductively from Jazz chord sequences using Bayesian models and computational simulations.

The theoretical part of the thesis presents how questions about learnability can be studied in a unified framework involving music analysis, cognitive modeling, Bayesian statistics, and computational simulations. A new grammar formalism, called Probabilistic Abstract Context-Free Grammar (PACFG), is proposed that allows for flexible probability models which facilitate the grammar-induction experiments of this study. PACFG can jointly model multiple musical dimensions such as harmony and rhythm, and can use coordinate ascent variational inference for grammar learning.

The empirical part of the thesis reports supervised and unsupervised grammar-learning experiments. To train and evaluate grammar models, a ground-truth dataset of hierarchical analyses of complete Jazz standards, called the Jazz Harmony Treebank (JHT), was created. The supervised grammar-learning experiments, in which grammars for Jazz harmony are learned from the JHT analyses, show that jointly modeling harmony and rhythm significantly improves the grammar models' prediction of the ground truth. The performance and robustness of the grammars are further improved by a transpositionally invariant parameterization of rule probabilities. Following the supervised grammar learning, unsupervised grammar learning was performed by inducing harmony grammars merely from Jazz chord sequences, without the observation of the JHT trees. The results show that the best induced grammar performs similarly well as the best supervised grammar. In particular, the goal-directedness of functional harmony does not need to be assumed a priori, but can be learned without usage of music-specific prior knowledge.

The findings of this thesis show that general prior knowledge enables an ideal learner to acquire abstract musical principles by statistical learning. In conclusion, it is plausible that much aspects of musical grammar have been learned by Jazz musicians and listeners, instead

of being innate predispositions or explicitly taught concepts.
This thesis is moreover embedded into the context of empirical music research and digital humanities. Current studies either describe complex musical structures qualitatively or investigate simpler aspects quantitatively. The computational models developed in this thesis demonstrate that deep insights into music and statistical analyses are not mutually exclusive. They enable a new kind of data-driven music theory and musicology, for instance through comparative analyses of musical grammar for different styles such as Jazz, Rock, and Western classical music.

Keywords: Music cognition, Computational cognitive science, Computational musicology, Bayesian statistics, Probabilistic machine learning, Artificial intelligence, Automatic music analysis, Jazz harmony

Zusammenfassung

Musikalische Grammatik beschreibt eine Gesamtheit von Prinzipien, die zum Verständnis sowie zur Interpretation der Struktur eines Musikstücks im Bezug auf einen Musikstil verwendet werden. Das Hauptthema der vorliegenden Arbeit ist die Grammatikinduktion der Harmonik — der Prozess des Erlernens struktureller Prinzipien aus der Beobachtung von Akkordfolgen. Die Frage, wie Grammatiken aus sequentiellen Daten erlernt werden können, ist eine Instanz einer zentralen Frage der Kognitionswissenschaft, wie aus der Beobachtung von Daten induktiv auf abstraktes Wissen geschlossen werden kann. Unter der Annahme, dass menschliches Lernen die Prinzipien rationalen Denkens respektiert, können Bayes'sche Kognitionsmodelle zur Simulation von Lernprozessen verwendet werden. In der vorliegenden Arbeit wird untersucht, welches Vorwissen es ermöglicht musikalische Grammatik induktiv aus Jazz-Akkordfolgen mit Hilfe von Bayes'schen Computermodellen und -simulationen zu lernen.

Der theoretische Teil der Arbeit legt dar, wie Fragen der Lernbarkeit in einem einheitlichen Rahmen durch Musikanalyse, kognitive Modellierung, Bayes'sche Statistik und Computersimulationen untersucht werden können. Ein neuer Grammatikformalismus, *Probabilistic Abstract Context-Free Grammar* (PACFG), wird präsentiert, der die Definition flexibler Wahrscheinlichkeitsmodelle ermöglicht, die in den Grammatikinduktionsexperimenten dieser Studie Verwendung finden. PACFG kann mehrere musikalische Dimensionen wie Harmonie und Rhythmus im Zusammenhang modellieren und *Coordinate Ascent Variational Inference* als Methode für das Grammatiklernen verwenden.

Der empirische Teil der Arbeit präsentiert überwachte (supervised) und unüberwachte (unsupervised) Grammatiklernsimulationen. Um die Grammatikmodelle zu trainieren und zu evaluieren, wurde ein Datensatz hierarchischer Analysen vollständiger Jazz-Standards, genannt *Jazz Harmony Treebank* (JHT), erstellt. Die überwachten Grammatik-Lernexperimente, in denen Grammatiken der Jazzharmonik aus den JHT-Analysen gelernt werden, zeigen, dass die gemeinsame Modellierung von Harmonie und Rhythmus die Vorhersagekraft der Grammatikmodelle signifikant verbessert. Diese Vorhersagekraft und die Robustheit der Grammatiken wird durch eine transpositionsinvariante Parametrisierung der Regelwahrscheinlichkeiten weiter verbessert. Im Anschluss an das überwachte Grammatiklernen wurde das unüberwachte Grammatiklernen durchgeführt, indem Harmoniegrammatiken lediglich aus Jazz-Akkordfolgen induziert wurden, ohne die JHT-Bäume zu beobachten. Die Vorhersagekraft der besten dabei induzierten Grammatik war ähnlich gut wie die beste Grammatik, die durch überwachtes Lernen erzeugt wurde. Insbesondere die Zielgerichtetheit der funktionalen Harmonik muss nicht a priori vorausgesetzt werden, sondern kann ohne Verwendung musikspezifischer Vorkenntnisse erlernt werden.

Die Ergebnisse dieser Arbeit zeigen, dass allgemeines Vorwissen einen idealisierten Lerner dazu befähigt sich durch statistisches Lernen abstrakte musikalische Prinzipien anzueignen. Zusammenfassend ist es somit plausibel, dass viele Aspekte musikalischer Grammatik von Jazzmusikern und -hörern gelernt werden, anstatt angeborene Veranlagungen oder explizit gelehrte Konzepte zu sein.

Darüber hinaus ist die vorliegende Arbeit in den Kontext der empirischen Musikforschung und der digitalen Geisteswissenschaften eingebettet. Aktuelle Studien beschreiben entweder komplexe musikalische Strukturen qualitativ oder untersuchen einfacher strukturierte Aspekte quantitativ. Die in dieser Arbeit entwickelten Modelle zeigen, dass sich tiefe Einsichten in die Musik und statistische Analysen nicht gegenseitig ausschließen. Sie ermöglichen eine neue Art datenbasierter Musiktheorie und Musikwissenschaft, beispielsweise durch vergleichende Analysen musikalischer Grammatiken verschiedener Stile wie Jazz, Rock und westlicher klassischer Musik.

Contents

Introduction

Music is a construct of the mind (Longuet-Higgins, 1979; Wiggins et al., 2010), it becomes alive when it is thought, played, or perceived. The study of music therefore benefits from taking into account the relation between the music and musicians, listeners, and dancers. Two motivating questions for the research presented in this thesis are how Jazz musicians think about the structure of the music they play in improvisations and how they learn to think in such ways. "To think" here refers to processes of the mind that take place subconsciously, not to conscious reflection of the musician. These question are particularly interesting, because improvisation is a highly complex cognitive task (Kenny and Gellrich, 2002). In jam sessions, musicians collectively coordinate their improvisation with ease, even when they never met before. This suggests the existence of abstract knowledge that musicians use to guide their play and freely explore a musical style. Such abstract knowledge about the structure of Jazz music is at least partly acquired implicitly by transcribing and playing the great masters — Thelonious Monk, Art Blakey, John Coltrane, or Charles Mingus. Musicians are thus unconscious about parts of their musical knowledge.

Musical grammar describes a set of principles that are used to understand the structure of a piece according to a musical style. In this understanding, the function of the grammar is not to normatively distinguish right from wrong. Instead, grammar is used to interpret musical structure (Steedman, 1996; Rohrmeier and Pearce, 2018). The grammar of Jazz harmony describes the structure of chord sequences and their interaction with other musical dimensions such as rhythm, form, and voice leading (Rohrmeier, 2020a). There are aspects of the grammar that are domain-general — aspects that are shared with other systems such as grammars of natural languages like English. For example, the principle of grouping certain words of a sentence or chords of a tune into constituents can be assumed to be domain-general. Other aspects are specific to music or even specific to a musical style. Statistical regularities of which notes or chords occur next to each other are for example style-specific (e.g., Saffran et al., 1999; see Rohrmeier and Rebuschat, 2012, for a review).

The main topic of this thesis is grammar induction for harmony — the process of learning structural principles from the observation of chord sequences. The question how grammars are learnable by induction from sequential data is an instance of the more general question how abstract knowledge is inducible from the observation of data — a central question of cognitive science. The majority of research about grammar induction has focused on natural language. The complexity of grammar acquisition lead Chomsky (1965, 1980, 1986) to the development of the idea of a *universal grammar*, an innate schematism of grammars for language that a human is able to learn. A current debate revolves around the Poverty of the Stimulus (PoS) argument that many abstract grammatical principles are not acquired through experience but are innate (e.g., Hauser et al., 2002; Jackendoff and Pinker, 2005; Berwick et al., 2011; Lewis and Elman, 2001; Pullum and Scholz, 2002; Perfors et al., 2011). It is the subject of ongoing research which principles are plausible to be learned through exposure, which are likely to be innate, and if the innate predispositions are specific to language. For music, less is known about grammar induction (Tsushima et al., 2018), and the question whether innate predispositions specific to music are required to acquire musical grammar remains unexplored. This thesis studies the theoretical requirements to acquire musical grammar in order to investigate whether innate predisposition specific to music is required.

Under the assumption that human learning follows approximately the principles of rational reasoning, Bayesian models of cognition can be used to simulate how the human mind builds rich and abstract models of the world to go beyond the data of experience (Chater et al., 2006; Griffiths et al., 2008; Tenenbaum et al., 2011), and musical grammar is an example of such an abstract model. Specifically, this study investigates what prior knowledge makes it possible to learn musical grammar inductively from Jazz music using Bayesian models and computational simulations. In particular, it is studied which style-general aspects of the grammar are learnable.

The present thesis is a continuation of previous research that applied formal grammar models to analyse musical structure. Following up on the formalizations of generative grammar by Chomsky (1965) and hierarchical music analysis by Schenker (1935), scholars started to apply generative grammar models to music in the 1960s and 1970s (Winograd, 1968; Baroni and Jacoboni, 1975; Sundberg and Lindblom, 1976). Lerdahl and Jackendoff (1983) later proposed the Generative Theory of Tonal Music (GTTM) to link hierarchical music analysis with music cognition, which influenced much following research (see e.g., Giblin, 2008; Bigand et al., 2009). Many recent grammar models, including the ones presented in this thesis, are probabilistic (McCormack, 1996; Gilbert and Conklin, 2007; Abdallah and Gold, 2014; Granroth-Wilding and Steedman, 2014; Quick, 2016; Tsushima et al., 2020). The approach used in this thesis is largely based on the grammar models proposed by Rohrmeier (2011, 2020a) and Rohrmeier and Neuwirth (2015) which were earlier also implemented and applied to model harmonic similarity (De Haas et al., 2009) or to improve automatic chord recognition (De Haas, 2012; De Haas et al., 2012). This thesis extends those approaches with a Bayesian probabilistic model and uses both supervised and unsupervised learning to infer probabilities of grammar rules from data.

From the standpoint of musicology, this study can be considered an empirical investigation of music-theoretical insights. For example, the interaction of harmony and rhythm and the goal-directedness of functional harmony are shown to significantly improve the learnability of a grammar for Jazz harmony. This study also contributes to the interaction of formal language theory and Bayesian statistics by proposing a novel grammar formalism that allows for flexible probabilistic modeling. The utility of this grammar formalism is demonstrated by the computational learning simulations. In particular, a link between the theories of Bayesian grammar inference (Kurihara and Sato, 2004) and semiring parsing (Goodman, 1999, 1998) is presented that aids the development as well as the implementation of probabilistic grammar models.

The thread of this thesis is aimed to comprehensibly lead from music theory and the interpretation of probability over formal language theory and Bayesian inference to computational experiments in music cognition. The text is intended to be written in a way such that detailed discussions corresponding to one of those fields are not necessary to understand the general argument. Most chapters begin with a small introduction and a summary that acts as an interface to the other chapters and the argument of this thesis. The thesis is organized hierarchically into parts, chapters, and sections. The first part introduces the concepts and assumptions this study is based on, presents its argument, and relates it to previous research. The second part gives a detailed description of the data, formalisms, and methods used in the thesis. It dives deep into the mathematical details of formal grammar models and Bayesian statistics. The third part of the thesis reports computational experiments for musical grammar learning. It includes supervised experiments in which grammars are learned from the observation of music analyses by human experts, as well as unsupervised experiments in which grammars are learned from chord sequences.

Chapter 1 starts the first part by introducing the musical structures considered in this study from the view of music theory. Chapter 2 presents the main research questions in the context of computational cognitive science in general and music cognition in particular (Pearce and Rohrmeier, 2012; Jackendoff and Lerdahl, 2006; Seifert, 1993). The Bayesian interpretation of probability is described in detail and used to link cognitive science, music theory, and computational modeling. Chapter 3 concludes the first part by comparing the present approach to related computational research on harmony.

The second part starts by presenting the Jazz Harmony Treebank (JHT) and its creation procedure in Chapter 4. The JHT is a dataset of tree-structured harmonic analyses by music-theory experts including the author. The treebank is used to train and to evaluate the grammar models presented in the computational experiments in the third part. Chapter 5 motivates and proposes Probabilistic Abstract Context-Free Grammars (PACFGs), a novel grammar formalism which enables the definition of complex probability models for learning musical grammars. Chapter 6 describes parsing — the process of inferring the grammatical structure of a sequence — for PACFGs as a generalization of parsing for context-free grammars in the semiring-parsing framework (Goodman, 1999, 1998). The last chapter of the third part (Chapter 7) proposes a flexible parameterization of PACFGs for grammar learning and describes the details of deriving a variational Bayesian inference method for PACFGs.

The third part of the thesis is empirical, it describes the computational experiments. The concrete grammar models used in the experiments are described in Chapter 8. Chapter 9 reports supervised learning experiments to demonstrate that grammar models profit from jointly modeling harmony and rhythm, and from transpositional invariance. Chapter 10 reports two unsupervised grammar-learning experiments to demonstrate that prior knowledge about the goal-directedness of functional harmony together with a prior preference for simple rhythms enables the induction of an interpretable harmony grammar of good quality. The final experiment is reported in Chapter 11. It shows how the abstract concept of goal-directedness of functional harmony can be learned from the observation of chord sequences and domain-general prior knowledge. The thesis finally concludes with a summary of the concrete contributions as well as a reflection and interpretation of the empirical findings in Chapter 12.

Parts of the work presented in this thesis have been already published in peer-reviewed conference proceedings (Harasim et al., 2018, 2019b, 2020a). Those articles can be found in the appendix. The thesis goes beyond that work, in particular by a detailed argumentation, an extensive literature review, a rigorous mathematical derivation of the methodology, and the presentation of novel computational experiments and results.

Theory and Background Part I

1 Hierarchical structures in Jazz harmony

The hierarchal organization of Western music—of its melodies, chord progressions, and rhythms—led to the development of sophisticated theories about musical structure (Schenker, 1935; Salzer, 1952; Cadwallader and Gagné, 2007; Larson, 1998; Heyer, 2012), on which formal and computational models are based (Lerdahl and Jackendoff, 1983; Steedman, 1996; Gilbert and Conklin, 2007; Rohrmeier, 2020a). This study builds on those theories and models to investigate how structural principles are learnable from sequential data. The existence of hierarchies that form the high-level organization of pieces from Western music is taken as an assumption based on music theory. There is an ongoing debate about how many hierarchical levels are relevant for music perception and if the hierarchical structure is caused by recursion (Temperley, 2011; Rohrmeier, 2013; Rohrmeier et al., 2015), but the existence of hierarchical structures is not seriously contested in music theory.

This chapter introduces the musical concepts and structures used throughout the thesis. It starts from the viewpoint of music theory by considering example analyses before further formalization is introduced in the following chapters. The approach presented in this chapter is based on the musical grammar models proposed by Rohrmeier (2011, 2020a), and Rohrmeier and Neuwirth (2015). Chapter 3 embeds the approach into the literature by comparing it to related research.

The Jazz standard *Take the "A" train* is taken as a running example in this chapter to illustrate various concepts. It is a good first example because of its simple harmonic and formal structure. Whenever necessary, it is complemented by more advanced examples such as the tunes *After you've gone* and *Afternoon in Paris.*

1.1 Jazz tunes as abstract musical entities

This study examines hierarchical structures in chord sequences of Jazz standards. The concept of a Jazz standard is abstract; in contrast to a composition from Western classical music in which all or most of the notes are given in a score, a Jazz standard is constituted by a basic

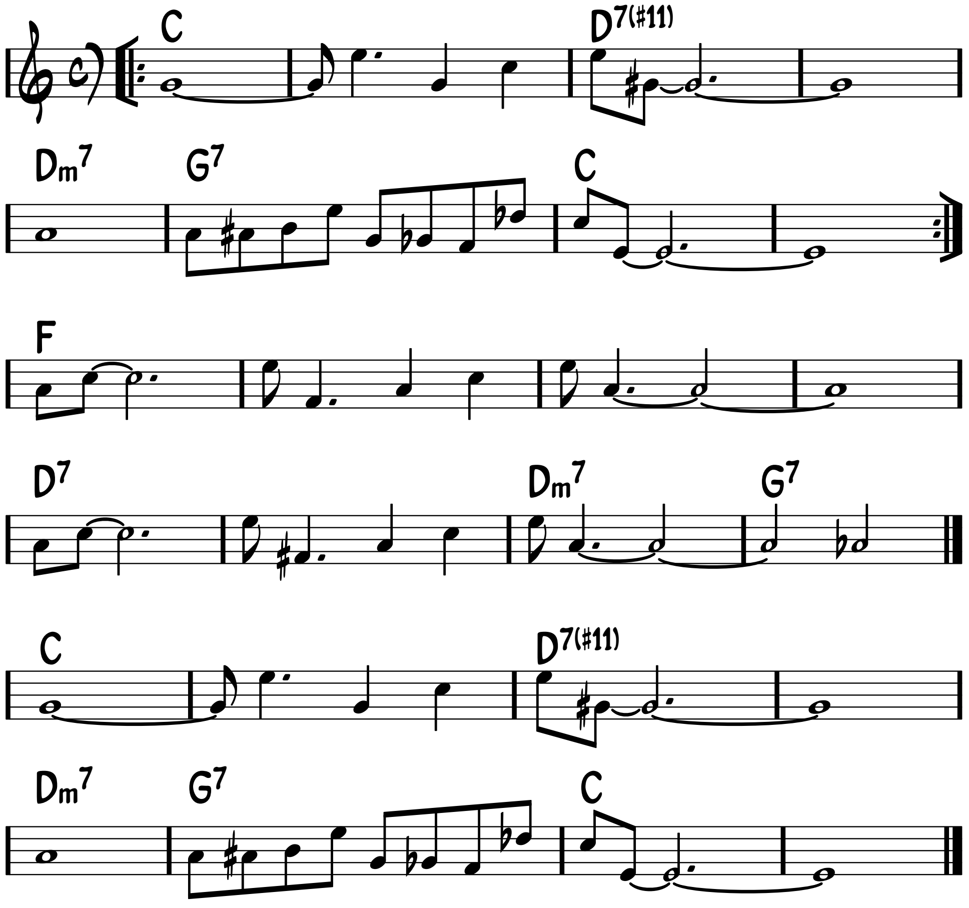


Figure 1.1 – Lead-sheet transcription of the Jazz standard *Take the A Train* in the key of C major. The transcription was created using LilyPond (http://lilypond.org/) with LilyJAZZ fonts (https://github.com/OpenLilyPondFonts/lilyjazz).

melody and a sketch of its harmonic accompaniment. The melodic and harmonic content of a Jazz standard is summarized in a *lead sheet*.[1] Both the melody and the accompaniment are usually modified and further elaborated in musical performances. Consider for example the Jazz standard *Take the "A" train*; a lead-sheet transcription is shown in Figure 1.1. The harmonic skeleton of the accompaniment is given by the chord symbols over the melody. The chord symbols C, $D^{7\sharp 11}$, Dm^7, and G^7, denote a major triad with root C, a dominant-seventh chord with sharpened 11 (the note G♯) and root D, a minor-seventh chord with root D, and a dominant-seventh chord with root G, respectively. For a detailed description of Jazz-chord symbols see for instance Levine (1990, 1995) or Sikora (2003).

For the purpose of this study, a *chord* is considered an abstract entity which is denoted by a chord symbol that is characterized by a root and a chord form. This representation abstracts for example from the temporal realization of a chord's notes as well as their register and ornamentation. Consider for instance the chords C and F in measure 1 and 9 of the lead sheet for *Take the "A" train* shown in Figure 1.1. Those two chords share the same chord form — they are both major triads — but are based on different root notes, C and F. In contrast, the chords $D^{7\sharp 11}$ and Dm^7 in measure 3 and 5 are an example of two chords that share the same root note but have different chord forms. The internal structures of chords, the relations between individual notes, are not further considered in this study. Chords' internal structures are a research topic of neo-Riemannian theory (Cohn, 1996, 1997, 1998; Douthett and Steinbach, 1998; Gollin, 2000), mathematical scale theory (Harasim et al., 2020b; Clough and Douthett, 1991; Carey and Clampitt, 1989; Clough and Myerson, 1985; Agmon, 1989; Domínguez et al., 2007; Harasim et al., 2019a) and geometric approaches to harmony (Harasim et al., 2016; Tymoczko, 2011; Callender et al., 2008; Tymoczko, 2006). This study rather considers chords as building blocks and focuses on the relations between the chords of a sequence.

Different performances of the same Jazz standard can be rather diverse, they only need to share the melodic and harmonic core that characterizes the Jazz standard.[2] To highlight the difference between a detailed composition or improvisation and music which is identified by more abstract properties, the former is called a *musical piece* while the latter is called a *tune*. Chord sequences of Jazz tunes are called *Jazz chord sequences*. Note that with this terminology, a Jazz standard itself is a tune while a recorded performance or a big-band arrangement of a Jazz standard is a musical piece. See for example Smither (2020) for a detailed discussion about the ontological status of Jazz tunes.

Since lead sheets are rough summaries of a single performance or a set of performances, there does not exist a definite lead sheet for any tune. Instead, Jazz lead sheets are compiled into *real books*, which provide divergent lead sheets for some Jazz standards (Kernfeld, 2006; Lovell, 2007; Smither, 2020). As shown exemplarily in the next section, different harmonizations of

[1] Lead sheets are also called fake sheets by other theorists and musicians.

[2] A recording of *Take the "A" train* by the Duke Ellington band can be found here: https://www.youtube.com/watch?v=cb2w2m1JmCY. A more recent interpretation by the singer Nikki Yanofsky can be found here: https://www.youtube.com/watch?v=K90xXn35d7o. These two very different interpretations of the same tune show the broad notion of the concept *tune*.

the same tune are, however, often similar. For simplicity, this study therefore considers only one lead sheet for each Jazz standard, which does not affect the presented argumentation.

1.2 Hierarchy in Jazz chord sequences

There are a music-theoretical and music-psychological arguments of various strength for the existence of hierarchical structures in music. We start this study with an elementary and theory-agnostic argument for hierarchical structure in Jazz harmony using the example of the tune *After you've gone* composed by Henry Creamer and Turner Layton in 1918. Afterwards, the observations are embedded into a more principled music-theoretical understanding.

According to the Oxford English Dictionary, a hierarchy is "a body of persons or things ranked in grades, orders, or classes, one above another" (Hierarchy, 2020). This study follows the proposal by Rohrmeier (2020a) and ranks chords in terms of their importance for the overarching coherence of the chord sequence. A chord is consequently lower ranked if it can be substituted or omitted in performances. The hierarchical organization of Jazz chord sequences is therefore reflected in the fact that musicians commonly substitute, omit, or insert chords at particular positions. In contrast, chords which are played in all or most of a tune's performances are likely to be important for the coherence of the whole sequence. It is shown in the next section how interchangeable or omittable chords can be understood as subordinations of higher-ranked chords.

The following paragraphs sketch a hierarchical organization of a chord sequence from the beginning of the Jazz standard *After you've gone* by observing four performances in which different chords are played. Figure 1.2 shows harmonic transcriptions of four performances of the tune. The four performances feature the musicians 1) Ella Fitzgerald, 2) Django Reinhardt, 3) the Huggee Swing Band, and 4) Jamie Cullum. The transcriptions were made by the author from recordings which are available on YouTube.[3] The transcribed harmonizations are transposed to C major for comparability; The keys of the recordings are E♭, G, D♭, and G major, respectively.

The chord sequences (2), (3), and (4) are compared to the version (1) which is taken as reference. Version (1) is also the version printed in *The New Real Book Volume 2* (Sher, 1991). The chord sequences (1) and (2) are very similar, they only differ in measures 2 and 4. Version (1) elaborates (2) by inserting the chord B♭^{7} in measure 2 and Em^7 in measure 4. Version (3) differs from (1) in three measures, in measures 2 and 4 as version (2) and additionally in measure 3: The chords Fm^6 and B♭^{7} are omitted in measure 2 to which F$^\triangle$ from measure 1 is extended. The chord Em^7 from measure 4 replaced C$^\triangle$ in measure 3 and A^7 is substituted by E♭$^{\circ 7}$. Version (4) differs from (1) in measures 2, 4, 5, and 6. In measure 2, the chord Fm^6 is

[3] (1: Ella Fitzgerald) https://www.youtube.com/watch?v=gCoVjIvkOEE
(2: Django Reinhardt) https://www.youtube.com/watch?v=BTH_Nn_TtDI
(3: Huggee Swing Band) https://www.youtube.com/watch?v=ew4-xVBcrmQ
(4: Jamie Cullum) https://www.youtube.com/watch?v=Sx-0_t8FMIE

(1)	$F^{\triangle}$	Fm^6 $B\flat^7$	$C^{\triangle}$	Em^7 A^7
	D^7	G^7	$C^{\triangle}$	$C^{\triangle}$
(2)	$F^{\triangle}$	Fm^6	$C^{\triangle}$	A^7
	D^7	G^7	$C^{\triangle}$	$C^{\triangle}$
(3)	$F^{\triangle}$	$F^{\triangle}$	Em^7	$E\flat^{\circ 7}$
	Dm^7	G^7	$C^{\triangle}$	$C^{\triangle}$
(4)	$F^{\triangle}$	$B\flat^7$	$C^{\triangle}$	D^7
	Dm^7 A^7	Dm^7 G^7	$C^{\triangle}$	$C^{\triangle}$

Figure 1.2 – Transcription of 4 harmonizations of the first 8 measures of the Jazz standard *After you've gone*, created by the author and transposed to the key of C major. The recordings on which the transcriptions are based are interpretations featuring (1) Ella Fitzgerald, (2) Django Reinhardt, (3) the Huggee Swing Band, and (4) Jamie Cullum. Triangles such as the one in the chord symbol $F^{\triangle}$ denote major-seventh chords. The chord symbol $E\flat^{\circ 7}$ denotes a diminished-seventh chord.

omitted. The chord D^7 from measure 5 replaced the chords Em^7 and A^7 in measure 4, making room for the chord sequence Dm^7 A^7 Dm^7 in measures 5 and 6.

The comparison of the transcriptions shows that the chords $F^{\triangle}$ in measure 1, G^7 in measure 6, and $C^{\triangle}$ in measures 7 and 8 are played at the same positions in all of the four transcribed performances (disregarding the displacement of G^7 in version (4)). It is therefore plausible that these three chords are most important for the coherence of the chord sequence, and thus ranked high in the harmonic hierarchy. The chords $C^{\triangle}$ in measure 3 and D^7 or Dm^7 in measure 5 are also mostly played and thus also hierarchically important. In contrast, the chords in measure 2 and 4 are different between the four versions, suggesting a lower ranking of those chords. In conclusion, the harmonic cornerstones of this chord sequence are the chords $F^{\triangle}$, G^7, and $C^{\triangle}$ in measures 1, 6, and 7, respectively. They function as pre-dominant, dominant, and tonic chords, respectively, as described in the next section.

1.3 Functional harmony

The definitions given in this and in the next section are based on the formal syntax models proposed by Rohrmeier (2011, 2020a), and Rohrmeier and Neuwirth (2015). The concept of *functional harmony* describes perceivable dependency structures between musical elements such as notes, chords, and keys. There are also non-functional relations between chords which are briefly discussed in Section 1.6. The core relations of functional harmony are *prolongation* and *preparation*. Prolongation expresses that two elements are an extension of a single higher-order element; Preparation is a relation between two elements in which the first element implies the second one, like a dominant chord implying a tonic chord. Consider for example the chords of the A part of *Take the "A" train* (measures 1 to 8), C $D^{7\sharp11}$ Dm^7 G^7 C. Figure 1.3 shows the harmonic dependency structure of this chord sequence. Directed arrows denote implications of preparations, and undirected arrows denote prolongations. The first C establishes the tonic and as such creates the expectation that C is reached again, which happens at the end of the phrase. This establishes a prolongation relation between the first and the last chord of the A part. The chord $D^{7\sharp11}$ functions as an applied dominant to G^7 and Dm^7 has subdominant function in C major. Both chords $D^{7\sharp11}$ and Dm^7 therefore prepare G^7. The chord G^7 functions as a dominant and therefore prepares the last tonic chord C. Abstractly, a chord X is said to *refer* to a chord Y if X either prolongs or prepares Y. Any direct dependency is thus understood as a harmonic reference. A graph that visualizes the harmonic dependency structure of a chord sequence such as the one shown in Figure 1.3 is called a *harmonic dependency graph* or *harmonic dependency structure*. In such graphs, prolongation and preparation are denoted by directed and undirected edges, respectively. The harmonic dependency graph of the whole tune *Take the "A" train* is shown in Figure 1.4.[4]

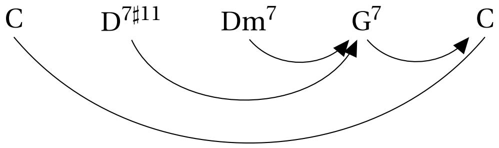


Figure 1.3 – Harmonic dependency structure of the A part of *Take the "A" train*. The undirected edge denotes the prolongation of the tonic chord C major. The directed edges denote preparations.

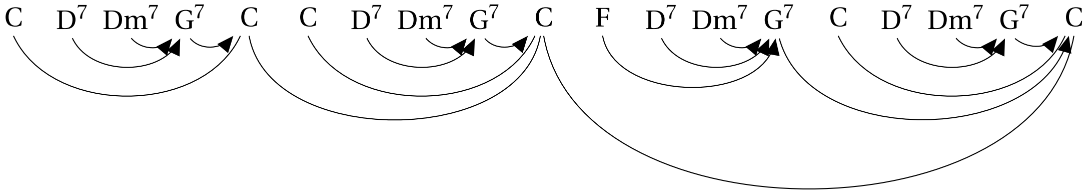


Figure 1.4 – Harmonic dependency structure of *Take the "A" train*. The chord symbol $D^{7\sharp11}$ is abbreviated as D^7.

[4] Note that dependency graphs are plotted with reversed edges in linguistics. There, the arrows show from which other word a word is generated.

Prolongation and preparation have an interpretation as a system for the build-up and release of harmonic tension (Rohrmeier, 2020a):

> [...] [T]he foundation of the build-up of musical tension lies in the set of (recursively nested) goal-driven implications [...], and the release of tension corresponds to every (sub)goal that is reached in a fulfilled preparation. Such a modeling of tension naturally implies that all tension is released when the final tonic is reached.

This implication-realization logic of harmonic dependencies acts as an interface between musical structure and expectancy (Rohrmeier, 2013). In contrast to the "timeless" harmonic dependency structure which describes a (part of a) tune as a whole, musical expectancy describes the temporal aspects of a listening experience. Since expectancy has been understood as fundamental for parts of musical experience, meaning, and emotion by Meyer (1956, 1967, 1973), it has grown to a central aspect of research on music perception (Huron, 2006). Hierarchical dependencies can describe aspects of musical expectancy as quoted above. Lerdahl and Jackendoff (1983), Jackendoff (1991), and Lerdahl (2001) use an alternative model to describe how non-local dependencies influence musical expectancy (see Section 3.2 and Rohrmeier (2020a) for a comparison). Most previous research, however, studied musical expectancy using models that take a local context of the music into account (Schmuckler, 1989; Narmour, 1990, 1991, 1992; Pearce, 2005a; Juslin and Vastfjall, 2008; Egermann et al., 2013; Sears et al., 2018; Pearce, 2018).

Harmonic dependency structures are assumed to be created implicitly by listeners to respond to the music (Jackendoff and Lerdahl, 2006). The denotation of one's perceived dependency structure of a chord sequence is an act of music analysis; it involves taking other musical dimensions into account such as harmonic rhythm, phrasing, musical form, and melody. For the analysis of whole tunes, the influence of form is most important as described in Section 1.5.

Sufficiently long chord sequences can be perceived in several ways, without one harmonic interpretation being clearly preferable. The purpose of a dependency structure is therefore to denote an individual and subjective understanding in an unambiguous formal representation. Even in the short example shown in Figure 1.3, one could argue that the chord Dm^7 is a (weak) prolongation of $\mathrm{D}^{7\sharp 11}$, because both chords share the same root note D. This alternative reading would be denoted by exchanging the edge from $\mathrm{D}^{7\sharp 11}$ to G^7 by an edge from $\mathrm{D}^{7\sharp 11}$ to Dm^7. It is further possible to emphasize that a C augmented triad is contained in the chord denoted by $\mathrm{D}^{7\sharp 11}$ in the lead sheet (the note E is present in the melody). That chord could then be analyzed as a preparation of Dm^7. Crucially, none of these three analyses can be considered the "correct" analysis of the tune. Instead, they denote different hearings that depend on particular performances. In what follows, only the first analysis shown in Figure 1.3 is considered further. In the opinion of the author, this analysis best describes the original recording of the tune by Duke Ellington and his band.

So far, this section considered the task of analyzing the dependency structure of a given chord sequence, similar to reductive analyses of more traditional harmony theory (Kostka and Payne, 1984; Strunk, 1979). There is a dual perspective to such analyses which describes the dependency structure by stepwise reconstruction of the sequence, starting from a single tonic chord. Stepwise reconstruction is also called *generation*. Figure 1.5 shows the generation steps of the A part of *Take the "A" train* which reconstruct the dependency structure shown in Figure 1.3. A fundamental principle of such generation is that chords must be generated adjacent to the chord they refer to. The order in which the chords are generated is therefore important. For example, the chord Dm^7 must be generated after the chord $\mathrm{D}^{7\sharp 11}$ to be adjacent to G^7. If otherwise Dm^7 would be generated first, then the order of $\mathrm{D}^{7\sharp 11}$ and Dm^7in the sequence would be reversed.

Harmonic dependency structures can be described more generally using the concept of scale degrees which abstract from the root a tune's key. A *scale degree* denotes a note on a diatonic scale relative to the tonic. It is written as a Roman numeral I, II, III, IV, V, VI, or VII, possibly with accidentals. In a C major scale for example, the note D is the scale degree II and the note G is the scale degree V. The canonical triads and seventh-chords of a scale, which are constructed by stacking of thirds, are also denoted by the scale degree of their roots. When the form of a chord is clear from the context, it can be omitted in scale-degree notation. The word scale degree is thus also used to refer to a whole chord.[5]

In cases in which the tonal center shifts from the tonic note to another note of the scale without constituting a proper modulation, scale degrees can be denoted relative to other scale degrees. A chord D^7which acts as a dominant applied to a chord G^7in a C major scale can for instance be denoted by V/V. With this notion of relative scale degree, a simple scale degree such as V could equivalently be notated as V/I to explicitly show the relation to the tonic. Using scale-degree notation, the chord sequence of the A part of *Take the "A" train*, C $\mathrm{D}^{7\sharp 11}$ Dm^7 G^7 C, can be written as I V/V II V I.

Scale degrees are mainly used for music-theoretical considerations in this study. The models used in the computational experiments are defined directly on chord symbols to minimize the domain knowledge put into the models. Minimal knowledge prior to the observation of musical data is desireable, because it simplifies the analysis of what is learnable from the observation of music — from listening.[6]

[5] Some authors who explicitly distinguish notes as scale degrees from chords as scale degrees use Arabic numerals with hats for the former and roman numerals for the latter. Since such a distinction is not crucial for this study, the introduction of the additional notation is avoided.

[6] If otherwise scale degrees and keys would be directly encoded into the models, sophisticated regularization methods would have to be used (Harasim et al., 2018).

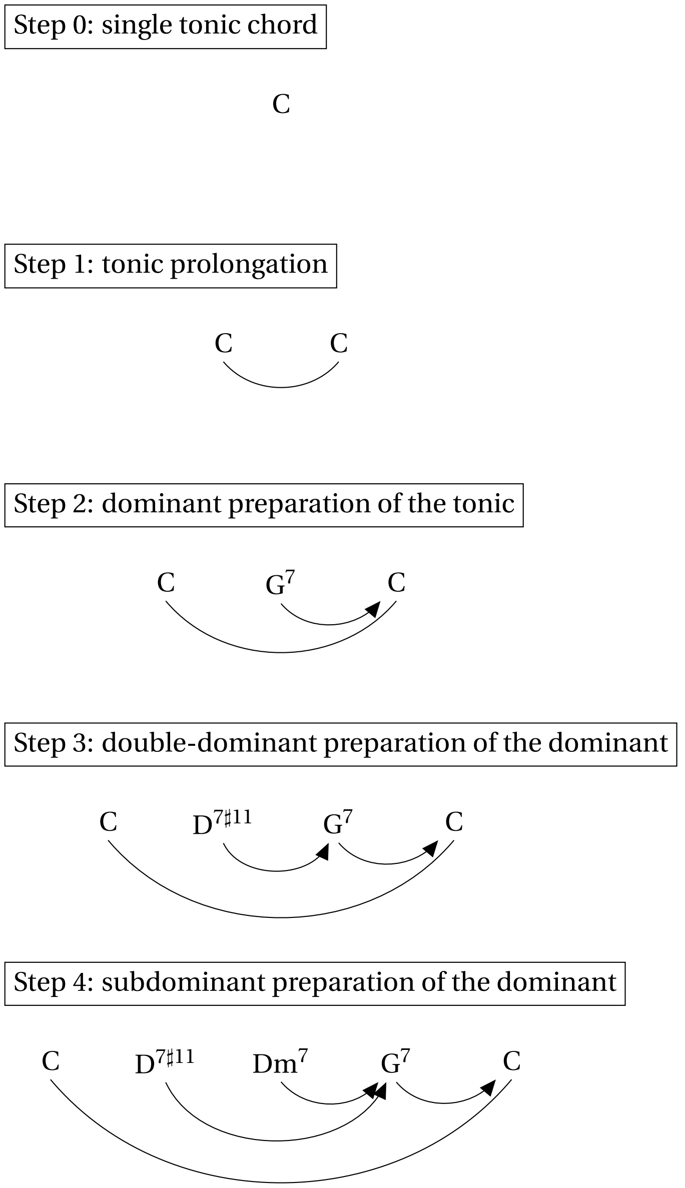

Step 0: single tonic chord
C
Step 1: tonic prolongation
C C
Step 2: dominant preparation of the tonic
C G7 C
Step 3: double-dominant preparation of the dominant
C D7♯11 G7 C
Step 4: subdominant preparation of the dominant
C D7♯11 Dm7 G7 C

Figure 1.5 – Stepwise generation of the harmonic dependency structure of the A part of *Take the "A" train.*

1.4 Prolongation and preparation as formal grammar rules

A system of rules that is able to generate all well-formed harmonic dependency structures is called a *generative grammar* for functional harmony. Such grammars differ for different styles and this study is mainly concerned with grammars for functional harmony of Jazz standards. Since it is hard to make clear distinctions between styles (Meyer, 1989), tonal Jazz is considered including Swing, Bossa Nova, Jazz Blues, Bebop, Cool Jazz, and Hard Bop, and excluding parts of traditional Blues, Modal Jazz, Free Jazz, and Modern Jazz. The core of the harmony grammars for these styles is furthermore expected to be similar to grammars for the harmonic structure of other tonal styles such as Baroque music (Rohrmeier, 2011).

The following definition presents a simple formalization of prolongation and preparation using context-free grammars, a standard formalism for modeling hierarchically structured sequential data (Manning and Schütze, 1999). Context-free grammars constitute a starting point for many generative models for the syntax of natural language in the Chomskian tradition (Chomsky, 1957, 1965, 1995; Adger, 2003).

Standard context-free grammar A *(standard) context-free grammar* $G = (T, N, \text{Start}, R)$ consists of a finite set T of *terminals*, a finite set N of *nonterminals* disjoint to T, an initial nonterminal $\text{Start} \in N$, and a finite set $R \subset N \times (T \uplus N)^*$ of *rewrite rules*, where $T \uplus N$ denotes the disjoint union and $(T \uplus N)^* = \biguplus_{n \in \mathbb{N}} (T \uplus N)^n$ denotes the set of all sequences (also called lists or strings) of terminals and nonterminals. A rule $(A, \alpha) \in R$ is denoted by $A \longrightarrow \alpha$ which reads as "the nonterminal A is rewritten into the sequence α". The grammar generates sequences of terminals by starting to apply a rule to rewrite the start symbol into a sequence of terminals and nonterminals. Afterwards, it iteratively generates rules to rewrite nonterminals of the sequence until the sequence only consists of terminals. Then, the process halts and the terminal sequence is returned.

Rohrmeier (2020a) formalizes a grammar of functional harmony with chord symbols as terminals, scale degrees as nonterminals, and the scale degree I as start symbol. This formalization assumes that each scale degree knows its key (Rohrmeier, 2011). The following paragraphs give a rather high-level description of the grammar rules. The grammar models used in the computational experiments are defined rigorously later in Chapter 8.

Preparation is formalized by rules of the form $X \longrightarrow Y \;\; X$ for scale degrees X and Y such that Y prepares X. The preparation of the first scale degree by the fifth scale degree is for example represented by the rule $\text{I} \longrightarrow \text{V} \;\; \text{I}$. The formalization further distinguishes two types of prolongation, *strong prolongation* and *weak prolongation*. The former describes a relation between chords with the same root and chord form while the latter more generally describes a relation between functionally equivalent chords (e.g., scale degrees II and IV). Strong prolongation is formalized by rules of the form $X \longrightarrow X \;\; X$ for scale degrees X (e.g., $\text{I} \longrightarrow \text{I} \;\; \text{I}$). Weak prolongation is formalized by rules of the form $X \longrightarrow Z \;\; X$ and $X \longrightarrow X \;\; Z$ for functionally equivalent scale degrees X and Z (e.g., $\text{I} \longrightarrow \text{I} \;\; \text{VI}$ in major or $\text{I} \longrightarrow \text{I} \;\; \text{III}$ in minor). Note that this

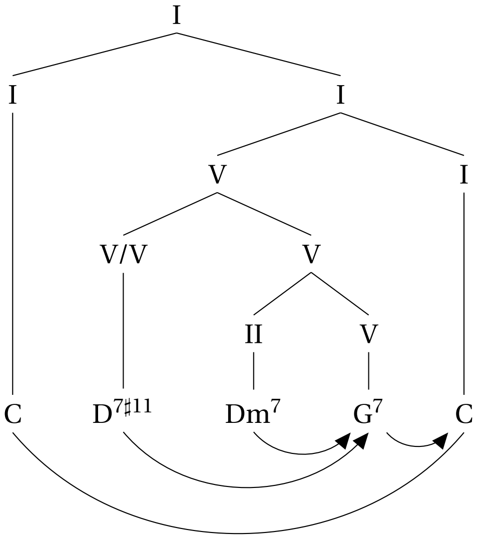


Figure 1.6 – Derivation tree of the A part of *Take the "A" train*. The harmonic dependency structure shown below the chord sequence stands in 1-to-1 relation to the derivation tree.

concept of weak prolongation is more general than in the *Generative Theory of Tonal Music* (GTTM) where prolonging chords are required to have the same root (Lerdahl and Jackendoff, 1983). In contrast to the GTTM, *departure* is not modeled as a primitive relation, because it is not consistent with the presented formalization of functional harmony (see Rohrmeier, 2020a).

Additional to grammar rules for prolongation and preparation, the grammar comprises unary terminal rules of the form $X \longrightarrow x$ for chord symbols x which are analyzable as scale degrees X (e.g., V ⟶ G^7 in C major). Furthermore, modulation can be formalized by unary rules which reinterpret any scale degree as a first scale degree in the modulated key. For example, the scale degrees II and IV in C major would be reinterpreted as scale degree I in D minor and F major, respectively.

A context-free grammar generates a sequence of terminals by recursive application of rules, starting from the start symbol. The A part of *Take the "A" train* can for example be derived by application of the binary rules

1) $\mathrm{I} \longrightarrow \mathrm{I}\ \mathrm{I}$,

2) $\mathrm{I} \longrightarrow \mathrm{V}\ \mathrm{I}$,

3) $\mathrm{V} \longrightarrow \mathrm{V/V}\ \mathrm{V}$, and

4) $\mathrm{V} \longrightarrow \mathrm{II}\ \mathrm{V}$,

together with the corresponding terminal rules. Each application of one of these binary rules

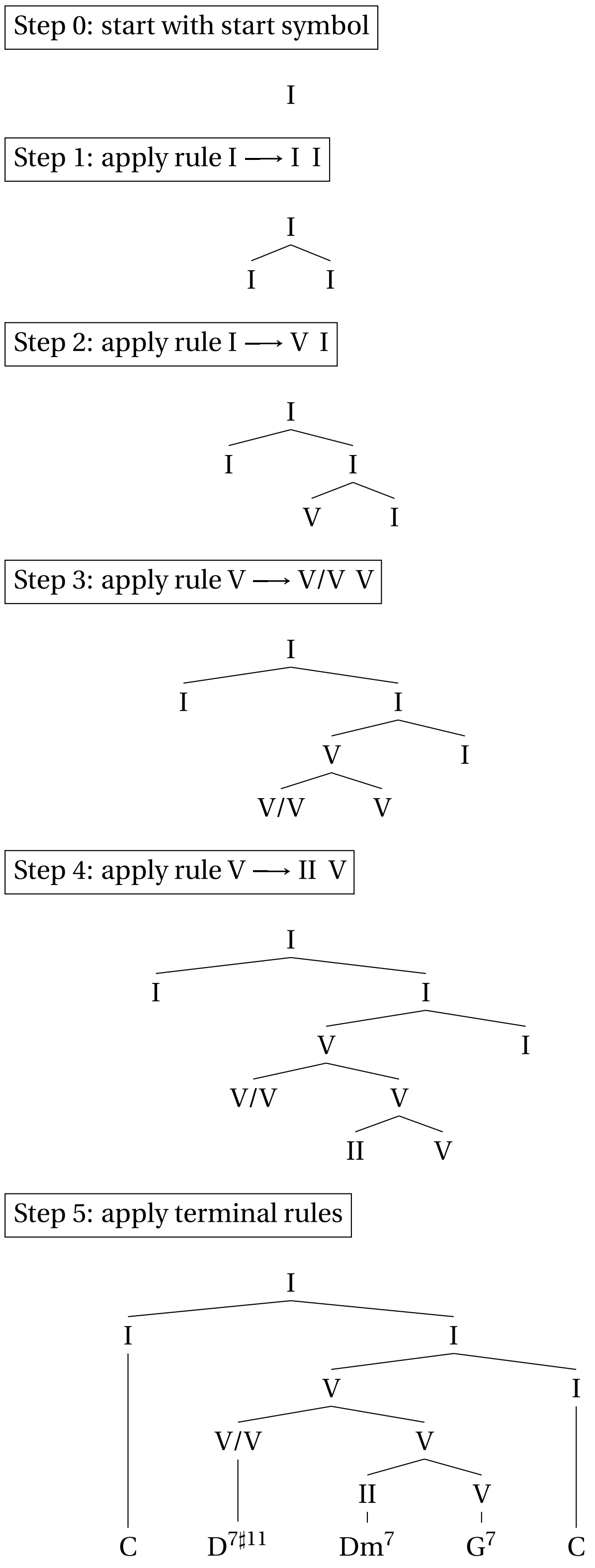

Step 0: start with start symbol
I
Step 1: apply rule I ⟶ I I
Step 2: apply rule I ⟶ V I
Step 3: apply rule V ⟶ V/V V
Step 4: apply rule V ⟶ II V
Step 5: apply terminal rules
V/V
II
C
D7♯11
Dm7
G7
C

Figure 1.7 – Stepwise generation of the derivation tree of the A part of *Take the "A" train.* All scale degrees are relative to C major. The generation steps 0 to 4 correspond to the generation of the harmonic dependency structure shown in Figure 1.5.

corresponds to one step in the generation shown in Figure 1.5, where the k-th rule corresponds to the k-th derivation step. The full generation process is shown as a *derivation tree* — also called *syntax tree* — in Figure 1.6. The harmonic reference graph shown below the chord symbols is not part of the derivation tree, but is closely related to it. The branches of the tree (from the root at the top to the leafs at the bottom) are generated using the four binary rules enumerated above. They therefore also correspond to the generation steps shown in Figure 1.5. To illustrate how to read the derivation tree, Figure 1.7 shows the mechanics of how the derivation tree is generated step by step. The stepwise generation of the tree does, however, not contain more information than the full derivation tree — the end product of the generation procedure.

The Jazz standard *Afternoon in Paris* is a good example for reinterpretation by modulation. A tree analysis of this tune's A part is shown in Figure 1.8. From the first to the last chord, the sequence modulates from C major to B♭ major to A♭ major and back to C major. The modulations are formalized by the unary rules $\mathrm{V}_{\mathrm{E}\flat} \longrightarrow \mathrm{I}_{\mathrm{B}\flat}$ and $\flat\mathrm{II}/\mathrm{V}_C \longrightarrow \mathrm{I}_{\mathrm{A}\flat}$. The first rule expresses that the chord B♭$^{\triangle}$ functions both as a tonic in B♭ major and a predominant in A♭ major. The second rule expresses that the chord A♭$^{\triangle}$ functions both as a tonic in A♭ major and a predominant in C major. Note that because of the goal-directedness of functional harmony,

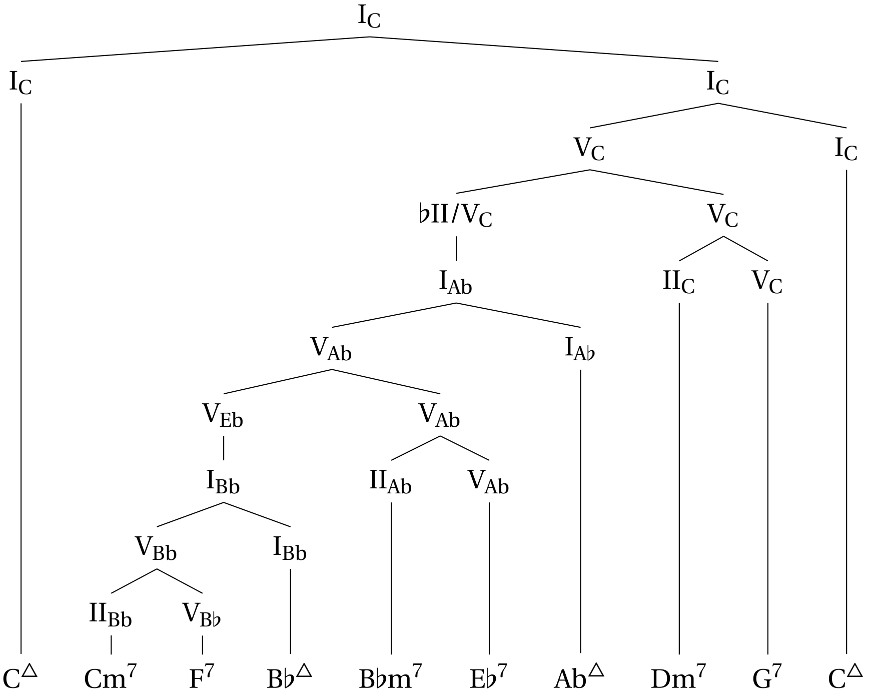


Figure 1.8 – Harmonic derivation tree of the A part of *Afternoon in Paris* in C major. The chord sequence modulates from C major to B♭ major, A♭ major, and back to C major. The local keys are indicated by subscripts. The modulations are represented by the unary rules $\flat\mathrm{II}/\mathrm{V}_C \longrightarrow \mathrm{I}_{\mathrm{A}\flat}$ and $\mathrm{V}_{\mathrm{E}\flat} \longrightarrow \mathrm{I}_{\mathrm{B}\flat}$. The first two chords constitute a parsimonious voice-leading connection which is not part of the syntactic structure.

the initial modulation of the sequence from C major to B♭ major is not directly represented by a single rule but as the composition of the modulations from C major to A♭ major and A♭ major to B♭ major.

1.5 Hierarchical phrase structure and form in Jazz standards

Harmonic analyses of whole Jazz standards benefit from the consideration of form (Rohrmeier, 2020a):

> [...] [H]armonic syntax characterizes harmonic dependencies and their interpretation. Form, by contrast, describes the regularities of phrases and their (hierarchical) organization as well as repetition structure in melodic, motivic or harmonic domains (Diergarten and Neuwirth, 2019). Hence, syntax and form serve different purposes, but they can interact in rich ways. They are closely linked when it comes to the understanding of prolongation and key structures that govern whole phrases and even entire pieces, and often, analyses of both harmonic syntax and form share large parts of their substructure.

Consider for example the complete derivation tree of *Take the "A" train* shown in Figure 1.9. The tune is structured into four phrases which constitute an AABA form. Each phrase corresponds to a subtree as indicated by the circled letters in the figure. In this example, the harmonic dependency structure is perfectly aligned with the formal structure, which is not always the case. Section 4.2 later proposes a solution to coordinate special kinds of misalignments between harmonic references and form such as it is the case for phrases which end in half cadences.

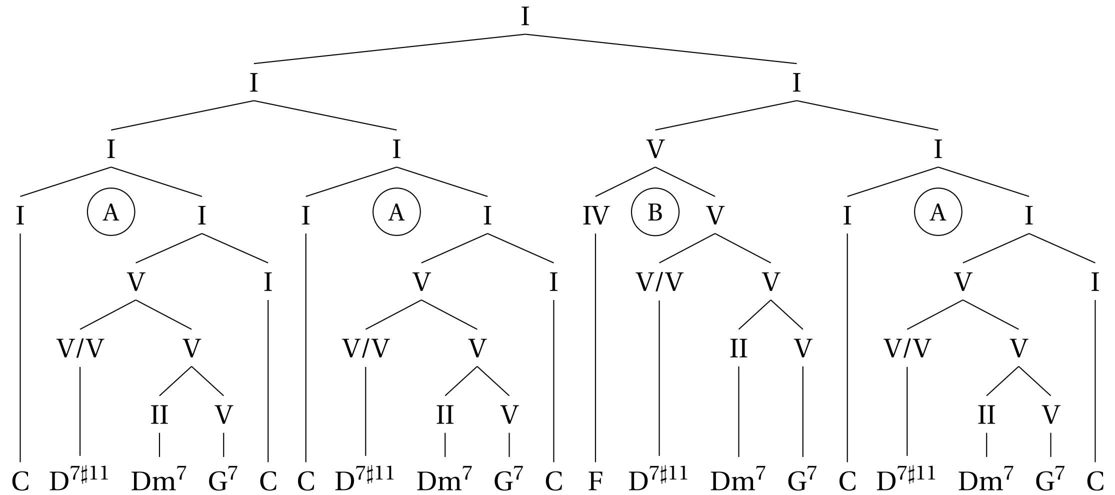


Figure 1.9 – Harmonic derivation tree of *Take the "A" train* in C major.The tune is structured into four parts AABA. Each part corresponds to a subtree as indicated by the circled letters.

The joint consideration of form and functional harmony has several benefits. It helps to resolve many ambiguities that would otherwise be hard to reason about. At the same time, it enables the formal description of structures which would otherwise be given in a more informal fashion. Consider for example the last chord of the first A part and the first chord of the second A part of *Take the "A" train*. Both chords are C major chords and as such are likely to constitute a direct prolongation from the viewpoint of pure harmonic reference relations. Yet they do not sound like they are a direct prolongation.[7] Instead, the last chord of the first A part sounds like an end while the first chord of the second A part sounds like a beginning. This is easily explained by consideration of the tune's phrases.

Musical form was studied in depth for Western classical music of the common-practice period (Schachter, 1980; Rothstein, 1989; Caplin, 1998; Cone, 1968; Cooper and Meyer, 1960). The concept of a phrase is there commonly also approached harmonically; one of the defining properties of a phrase is that it ends with some form of a cadence. Such harmonic phrases tend to be reinforced by surface features such as textual changes, motives, long notes, and rests. A similar approach was applied to Jazz tunes, which also identifies phrases by essential harmonic motions such as cadences (Forte, 1995). More recently, Love (2012) broadened the cadential requirement to more general forms of closure for the application of the phrase concept to improvised Jazz melodies.

Phrase structure in Jazz is strongly correlated with hypermeter. That is, with the regular organization of "groups of multiple measures that seem to begin with a relatively strong beat; two- and four-bar hypermeasures are ubiquitous in jazz" (Love, 2012). This is well illustrated by a quote by Strunk (1979):

> The rhythm of bop harmony at the foreground level is virtually always duple at each division or subdivision: the duration of most chords is two, four, six, or eight beats; phrases are two, four, six, or eight measures long; compositions [i.e., tunes] are usually twelve, sixteen, or thirty-two measures long. The utter simplicity and rigidity of these rhythmic structures highlights the complexity and subtlety of the jazz rhythmic nuances and syncopations which proliferate against the basic duple pulse.

Indeed, the running example *Take the "A" train* is 32 measures long, it consists of four 8-measure long phrases, and the duration of the chords is either 2, 4, or 8 beats (i.e., quarter notes). A corpus study by Salley and Shanahan (2016) confirms that this holds for many Jazz standards. Moss et al. (2020) make similar observations for Brazilian Choro, a musical style closely related to Jazz.

Despite the close relation between phrase structure and hypermeter, it is important to distinguish rhythm from meter (Lerdahl and Jackendoff, 1983; London, 2012). *Meter* can be conceptualized as a fixed grid, "an ongoing hierarchical temporal framework of beats aligned

[7] The described listening experience is the subjective listening experience of the author.

with the musical surface" (Jackendoff and Lerdahl, 2006). Chords are positioned in the metrical grid by their onset and offset. The "segmentation of the musical surface [the chords] into motives, phrases, and sections" is called the *grouping structure* of the chord progression (Jackendoff and Lerdahl, 2006), and its *rhythmic structure* can be understood as the relation between grouping and meter.

Rhythms become interesting when the grouping structure is not perfectly aligned with the metrical grid. Common examples for misalignments in Jazz are upbeats (also called pickup or anacrusis) and syncopation. Measure 4 in the second version of the A part of *After you've gone* shown in Figure 1.2 is an instance of a harmonic upbeat, because the A^7chord in that measure refers to a chord on a hypermetrically much stronger measure — the D^7chord in measure 5. Syncopation is often realized during the interpretation of a lead sheet, but commonly not explicitly written there.

In some Jazz standards, phrases are composed to form compound phrases. Such a nesting of phrases can occur multiple times, therefore giving rise to a hierarchical phrase structure. The large-scale structure of *Take the "A" train* can for example be considered a binary form consisting of two compound phrases, AA and BA. A second common form is the period-like form ABAC realized by tunes such as *All of me*, *How high the moon*, and *A fine romance*. The term *period-like* is used here to acknowledge both the similarities and the differences between the period form in Jazz and Western classical music. In both genres, periods consist of two phrases of which the first closes on a dominant chord and the second starts as the first and closes on a tonic chord. The cadences and the melodic movements are, for example, more freely interpreted in Jazz. However, a detailed comparison is beyond the scope of this study. Ternary forms AAB are also common, such as in *Song for my father*, *Blue monk*, and *Mr. P.C.*.

1.6 Non-functional harmonic relations between chords

The transition from the first to the second chord of *Afternoon in Paris*, shown in Figure 1.8, is an example of a relation between chords which is not analyzed as part of the functional harmonic structure. This relation is that of a *parsimonious voice leading*, that is a chord transition with a small voice movement (Harasim et al., 2016; Douthett and Steinbach, 1998; Cohn, 1997; Tymoczko, 2011). From $C^{\triangle}$ to Cm^7, the E and B move down by one semitone to E♭ and B♭, respectively, and C as well as G stay the same. Despite their importance for the coherence of the sequence (Wall et al., 2020; Huron, 2016), voice leading is not included in the tree analysis, because it is not directly considered a part of functional harmony. Moreover, if the chords would hypothetically be analyzed as being directly related, then only one of the two harmonic relations to the chords F^7 and $C^{\triangle}$ could be represented in a tree. Other examples of non-functional harmonic relations include linear chromatic or diatonic shifts (Rohrmeier, 2020a).

2 Reasoning and learning as probabilistic inference

This chapter presents the main research questions that motivate the research presented in this thesis and describes how they are studied using probability theory, Bayesian statistics, and computational learning experiments.

2.1 This study's research questions and their relation to music cognition and cognitive science

Jackendoff and Lerdahl (2006) propose studying the human capacity for music by disentangling 1) the cognitive structures invoked by music, 2) the musical grammar used to construct such structures, 3) the acquisition of musical grammar, and 4) the innate resources for music acquisition. The two core questions of this study consider the cognitive structures invoked by Jazz harmony and the acquisition of the corresponding grammar.

Q1: Which harmonic dependency structures are constructed by human minds in response to Jazz tunes?

Q2: What prior knowledge makes it possible to acquire musical grammar from an exposure to Jazz tunes?

The first question can be considered a cognitive framing of a music-theoretical problem. In general, humans create mental representations of perceptual input to learn about the environment and to make predictions about the future (Bubic, 2010; Pitt, 2020). Accordingly, humans create mental representations of what they hear when listening to music (Rohrmeier and Koelsch, 2012). Such representations might be created fully unconsciously or partly consciously. In both cases, they can be interpreted as music analyses — as listeners' *understanding* of the music (Jackendoff and Lerdahl, 2006) — whose sophistication depends to a large extent on the musical expertise of the individual listener. As structured representations are frequently viewed as theoretically central in cognitive science (Fodor, 1981; Chater et al., 2006), they are a connecting factor between cognitive science and music theory (Rohrmeier and Rebuschat, 2012). The second question is a fundamentally cognitive one since it concerns the learning process in which musical knowledge is acquired through the observation of examples. This study operationalizes the exposure to Jazz tunes as observations of chord sequences. The second question is moreover an instance of one of the main questions in cognitive science: "How does abstract knowledge guide learning and reasoning from sparse data?" (Tenenbaum et al., 2011)

2.1.1 Statistical learning and computational modeling

One fundamental assumption this study builds on is that much musical competence of both musicians and nonmusicians can be acquired through mere exposure — by conscious and unconscious listening to music (Rohrmeier and Rebuschat, 2012; Bigand and Poulin-Charronnat, 2006; Huron, 2012; Loui, 2012; Tillmann, 2005; Dienes and Perner, 1999; Huron, 2006). Such acquisition is commonly referred to as *implicit learning* or *statistical learning*. Implicit learning emphasizes the unconscious aspect, and statistical learning the quantitative aspect of learning. However, both terms are used interchangeably in the literature (Rohrmeier and Rebuschat, 2012). As a simple example of statistical learning, consider the behavioral experiment by Jonaitis and Saffran (2009) who found that listeners make use of statistical information when learning an artificially created musical style which allows for some chord transitions and disallows others. More precisely, they trained participants on chord-sequence examples from one of two styles represented by simple formal grammars.[1] They then found that participants were able to assign newly generated chord sequences to the style they were created from, significantly above chance level. Since the frequency of each chord was the same for both artificial styles, the results suggest that the participants based their decisions on chord transition probabilities.

The current debate in music cognition about which musical concepts can be acquired by statistical learning is another motivation of this study. It is rather a consensus that simple regularities such as frequent chunks of two or three notes or chords can be acquired by statistical learning (Rohrmeier and Rebuschat, 2012; Saffran et al., 1999; Tillmann and McAdams, 2004; Schön et al., 2008; Loui et al., 2009; Rohrmeier et al., 2011; Rohrmeier and Widdess, 2017). Rohrmeier and Cross (2014) further argue that implicit learning of grammatical structure is also plausible when only parts of sequences are well-formed. Rohrmeier and Cross (2013) argue that prior knowledge and processing constraints are important for successful implicit learning of artificial melodies. The possibility that abstract concepts such as structural properties of a context-free grammar can be acquired from mere exposure to sequential data is far less clear and subject of a current cognitive debate (Rohrmeier and Rebuschat, 2012). There is, however, growing empirical evidence that implicit learning can go beyond the learning of chunks and create knowledge about nonlocal relations in music (Kuhn and Dienes, 2005; Rohrmeier and Cross, 2009; Rohrmeier et al., 2012, 2014). Such empirical evidence is largely based on artificial grammar experiments which have the drawback that they isolate musical

[1] Regular grammars were used to create the chord sequences.

dimensions such as harmony and rhythm and use artificial stimuli instead of excerpts of real music. This can be problematic since music theory argues for the importance of interaction between musical dimensions (Rohrmeier and Rebuschat, 2012; Cadwallader and Gagné, 2007; Lerdahl and Jackendoff, 1983).

Computational models offer an approach to music cognition that is in some sense complementary to empirical psychology. Empirical psychology works directly with human participants and studies their percepts with quantitative methods. The laboratory nature of psychological experiments, however, requires stimuli that can be far away from the actual object of study — the music. Furthermore, abstract knowledge representations of human minds are hard to study in psychological experiments, because "[t]he difficulty with studying minds and brains is that they are very difficult to measure" (Wiggins et al., 2011). In contrast to empirical psychology, computational cognitive models can focus on abstract knowledge representation and are easily applied to actual music through corpus studies. They are specified as precisely as that they are implementable on a computer (Wiggins et al., 2011; Wiggins, 2011; Temperley, 2012), and explicitly represent the knowledge available prior to learning. Computational cognitive models are important to advance the understanding of the human mind, because they allow research to focus on the question *what* the mind does instead of *how* it does it. This distinction was most famously proposed by Marr (1982) and is discussed in greater detail to conclude this chapter in Section 2.6. The major drawback of computational cognitive models is that much argumentation and philosophical background is required to relate computational results to music perception. This is, however, not a problem from the view of cognitive science which understands itself as the interdisciplinary study of natural *and* artificial intelligence (Thagard, 2019).

Since there is to date no methodology available (and there might never be) to directly study abstract knowledge representation in the human mind, this study investigates the learnability of aspects of such representations in principle. This has the advantage that it can be studied using computational cognitive models and learning simulations as described in the following sections. For computational models of learning, a central question is what knowledge is available to the learner prior to the observations she learns from. This includes the form of possible knowledge representations, the space of possible hypotheses about the world, and prior preferences over hypotheses (e.g., a preference for simple hypotheses). The role of prior knowledge is essential to Bayesian models of cognition, because they assume that nothing can be learned without prior knowledge — the only possibility to learn is to update knowledge. In addition to the prior knowledge, the mechanics of the learning process must be modeled rigorously. It defines the interaction of the learner with the rest of the world.

As outlined at the beginning of this chapter, the main research questions of this study are which prior knowledge enables learning of grammars for harmony and which aspects of harmonic syntax are learnable. More concretely, the goal is to find domain-general or at least style-general prior knowledge that enables statistical learning of the characteristics of Jazz harmony and form in a way such that the results are interpretable from the view of music theory. Candidates for potentially important prior knowledge include the joint consideration of harmony and rhythm, relative pitch perception, and a preference for simple rhythm. Assuming such prior knowledge, Chapter 10 presents a model which learns a harmony grammar for Jazz standards that successfully uses nonlocal dependencies and musical form. In Chapter 11 furthermore presents a model that is able to learn the goal-directedness of Jazz harmony from minimal and domain-general prior knowledge.

2.1.2 Learnability arguments and counterarguments of abstract syntactic principles in natural language

Computer modeling is one of the three main approaches of cognitive science — along with experimental psychology and neuroscience (Temperley, 2012). Learnability arguments similar to the one presented in this study were previously made for domains other than music. For example, Kemp and Tenenbaum (2008) argue that abstract knowledge is learnable in the form of knowledge graphs for many domains such as biology, vision, and topography. Such computational arguments make use of what are called Bayesian models of cognition (Griffiths et al., 2008), which are discussed below in the following sections.

A similar learnability argument to the one presented in this study was proposed by Perfors et al. (2011) who used computational simulations to show the learnability of abstract syntactic principles in natural language. That argument is a contribution to the “Poverty of the Stimulus” (PoS) debate in linguistics. The PoS debate discusses the statement whether children are exposed to data rich enough to acquire all features of their first language without the need of language-specific predispositions. A prototypical argument against that statement is the following: Since children learn the correct syntactic constructions when they acquire their first language, even when they have no direct evidence for corner cases, their generalizations must be guided by abstract knowledge, by some inductive bias. The debate now revolves around the question whether that abstract knowledge is innate or learned, “nature versus nurture”. Since some part of the learning capability must be innate, the question is more precisely stated as to which extent the inductive bias is innate and whether that predisposition is specific to language (Chomsky, 1975; Piattelli-Palmarini, 1980; Hauser et al., 2002; Jackendoff and Pinker, 2005). Such language-specific predisposition is referred to as *universal grammar*. The similar question of whether there exists music-specific innate knowledge was later asked by Jackendoff and Lerdahl (2006).

The side that argues for the existence of innate language-specific predispositions (Chomsky, 1965, 1971, 1975; Crain and Nakayama, 1987; Crain, 1991; Legate and Yang, 2002; Berwick et al., 2011) considers for example the case of auxiliary fronting in English to construct questions from statements. For instance, the question *Is the man a musician?* is constructed in such a way from the statement *The man is a musician.* The argument then proceeds by the observation that there could at least two grammatical rules be learned from such examples: 1) the first (leftmost) auxiliary verb in the sentence moves to the front or 2) the predicate of the sentence moves to the front (which reflects hierarchical phrase structure). Both rules coincide in the above example but diverge for more complex statements such as *The man who is playing piano is Thelonious Monk.* The first rule results in the ungrammatical question *Is the man who playing the piano is Thelonious Monk?* while the second rule results in the correct question *Is the man who is playing the piano Thelonious Monk?* The core of this poverty of the stimulus argument is then that complex questions of that kind are non-existent in child-directed speech in sufficient quantity to make the correct inference. Therefore, hierarchical phrase structure is assumed to be innate.

The other side which argues against innate language-specific predispositions uses psychological experiments and computational simulations to accumulate evidence for the learnability of abstract syntactic principles (Gomez and Gerken, 1999; Lewis and Elman, 2001; Pullum and Scholz, 2002; Perfors et al., 2006, 2011). For instance, Lewis and Elman (2001) use recurrent neural networks applied to a corpus of child-directed speech (MacWhinney, 2000) to show that "Chomsky's poverty of stimulus argument that structure dependence must be a principle of UG [Universal Grammar] fails to hold once stochastic information is admitted". As a second example, Perfors et al. (2011) propose a Bayesian grammar-learning model which demonstrates that an ideal learner (also called agent in this study) could infer the hierarchical phrase structure of language from child-directed speech using domain-general prior knowledge.

Although particular linguistic examples such as auxiliary fronting do not translate 1-to-1 to music, the general question how abstract structural concepts are learnable does. By providing initial evidence that hierarchical phrase structure is also learnable in the case of Jazz standards without music-specific prior knowledge, this study contributes to the general argument for the power of domain-general prior knowledge in conjunction with statistical learning.

Note that the statement that aspects of musical grammar are learnable does not imply that listeners actually learned and use them for perception. Musical expertise varies a lot across individuals, possibly much more than language expertise. This variation is partly caused by the different levels of engagement with music in general and specific musical styles in particular. In contrast to language where nearly every individual consumes and produces large amounts of spoken language, detailed musical knowledge is only needed by musicians. It is plausible that Jazz musicians learned aspects of musical grammar that are learnable in computational simulations. Non-musicians who have little exposure to Jazz are much less likely to have learned any details of the grammar of Jazz. Psychological experiments in which random participants do not perceive hierarchical structures in music can therefore not necessarily be interpreted as evidence against musical grammar. For natural language, the authority of native speakers simplifies the situation. The concept of a native musician is, however, questionable.

2.2 An agent model for learning simulations

This study uses computational experiments to investigate the research questions introduced at the beginning of this chapter. That is, a scenario is considered in which an agent has some prior knowledge about Jazz music or music in general. The agent then learns implicitly by observing ("listening to" or "engaging with") Jazz chord sequences. What would be reasonable to assume how this agent analyzes the harmonic structure of a new, unfamiliar tune? Probability theory and Bayesian statistics provide a mathematical framework to study this question. The question how a human individual learns is thus shifted to the question how an idealized rational agent learns. Such agents are also referred to as ideal learners. The shift is an approximation that enables rigorous scientific reasoning. Importantly, individual differences are not ignored by this approximation but modeled by the prior knowledge that is brought into the learning process. Computational learning experiments can therefore also be considered a modern and quantitative derivative of traditional thought experiments.

Which harmonic reference structure an agent choses to analyze a Jazz chord sequence depends on three crucial factors:

Prior knowledge: what the agent knows before learning,

Learning mechanism: how the observation of one chord sequence changes the agents view on Jazz music, and

Data: which Jazz tunes are observed during learning.

Probability theory provides the tools to study the agent model including the uncertainties associated with unknowns. In particular, Bayesian models of cognition explicitly represent and coherently integrate the three factors into a uniform mathematical framework (Griffiths et al., 2008). Probability is in this framework interpreted as *plausibility* (Jaynes, 2003), *quantification of uncertainty* (Bishop, 2013), or *degree of belief* (Chater et al., 2006). The term plausibility is more common in contexts in which probability theory is used as extended logic. The terms quantification of uncertainty and degree of belief emphasize the reference to the rational agent. All three terms refer to what is known as the objective Bayesian interpretation of probability (Hájek, 2019). A probability is a number that represents the agent's degree of belief that a statement holds; it is objective in the sense that the agent is assumed to act in a rational manner — independent of subjective judgments, hopes, or fears. Moreover, probability theory can therefore be considered a natural extension of classical binary logic as described in the following paragraphs.

The axiomatic formalization of probability theory as an extended logic calculus originated from the economist Keynes (1929) and the mathematician Jeffreys (1939). They proposed to interpret probability as a degree of rational belief ranging between certainty and impossibility and considered classical deductive logic a special case that only considers the values *certain* and *impossible* (Cox, 1946). Keynes' and Jeffrey's interpretation of probability was mathematically justified by the physicist Cox (1946, 1961) who derived the laws of probability theory from the requirement of consistency with Boolean algebra and elementary postulates about common sense reasoning. Cox' argumentation is extended, for example, by Jaynes (2003) to derive Kolmogorov's calculus of probability — the axiomatic probability theory of modern mathematics (Kolmogorov, 1933). See for example Jaynes (1986) for more historic information.

The interpretation of probability as degree of belief is fundamentally different from the so called *frequentist* interpretation which defines the probability of an event as its relative frequency in an infinite sequence of independent trials. Cox (1946) writes about the relation between the interpretations:

> Probability is recognized also as providing a measure of the reasonable expectation of an event in a single trial. [...] According to the second main school of probability [i.e., the Bayesian interpretation], this measure of reasonable expectation, rather than the frequency in an ensemble [i.e., in an infinitely repeated process], is the primary meaning of probability.

The frequentist understanding of probability crucially relies on the possibility of indefinite identical repetition and models uncertainty as variance of such repetition. It is thus not as natural to apply to learning simulations as the Bayesian interpretation.

Fortunately, modern mathematics studies probability theory based on the axiomatic system proposed by Kolmogorov (1933). Since the theorems derived from Kolmogorov's axioms are independent from the interpretation of probability, they can be used with all sound interpretations. In contrast, methods of frequentist statistics such as classical hypothesis testing using p values rely on the frequentist interpretation. They are therefore not used in this study.

2.3 Interpretation of fundamental concepts of probability

This section presents fundamental concepts of probability together with their notation and Bayesian interpretation. The focus lies on describing the intuition; mathematical details are provided in footnotes. The motivation of this writing is that the fundamentals of modern probability theory are rarely presented together with their Bayesian interpretation in a concise and comprehensible manner. In fact, no exhaustive source that applied to the problem at hand was found in the literature. The work by Burgoyne (2012) is closely related, but he used a frequentist interpretation; Temperley (2007) gives a good introduction to the intuition behind Bayesian reasoning about discrete random variables. The aim of this chapter is to extend those approaches by presenting the foundations of measure-theoretic probability theory along with their Bayesian interpretation to construct state-of-the-art learning models step by step. The gap between modern probability theory and statistical applications as observed by Fine (1973) still exists in many disciplines — particularly in music cognition. Since this gap seems to be caused by confusion about the interpretation and construction of probabilities, the definition and interpretation of the mathematical objects used in this study are made explicit. It is, however, assumed that the reader is familiar with probability theory in some form.

Under the Bayesian interpretation, all probabilities represent a degree of belief of a rational agent who reasons about the aspects the world that she cannot observe or did not observe yet. For example, the grammar of Jazz harmony can never be observed directly and its peculiarity is therefore always associated with uncertainty. The same is true more generally for all abstract cultural entities. In statistics, such entities that are not directly observable are called *latent*. In contrast, examples of observable entities include Jazz chord sequences and harmonic reference analyses of any particular analyst. The agent can reason about observable entities based on prior knowledge and other observations, but she might also be able to observe them and acquire certain knowledge.

In probability theory, the agent's reasoning about the unknown aspects of the word is formalized as reasoning about statements about the world. Such a *statement* $A \subseteq \Omega$ is modeled as the set of worlds in which the statement holds, where Ω denotes the set of all possible worlds.[2] The set of all statements is denoted by $\mathscr{A}$.[3] The belief of the agent is modeled as a probability measure $\mathbb{P}$ which assigns each statement A a real number $0 \leq \mathbb{P}(A) \leq 1$ that represents the plausibility of A.[4] The values 0 and 1 are interpreted as impossibility and certainty, respectively. Note that an additional agent can be represented by a different probability measure, for example denoted by $\mathbb{Q}$.

Statements about the world commonly refer to certain aspects of the world modeled as functions $\check{x}\colon \Omega \to X$, where X is the set of possible values of $\check{x}$, called its *range*. Aspects of the world whose values are unknown are called *random variables*.[5] The uncertainty about the value of $\check{x}$ thus results from the uncertainty in the world. One common form of statements is that the value of a random variable $\check{x}$ is contained in a set $Y \subseteq X$. The probability of that statement is written as

$$\mathbb{P}(\check{x} \in Y) := \mathbb{P}(\{\omega \in \Omega \mid \check{x}(\omega) \in Y\}). \tag{2.1}$$

[2] Some researchers criticize the assumption that there is a single set of all possible worlds. This is one of the criticisms about the Bayesian interpretation of probability (Fine, 1973).

[3] The mathematical structure of $\mathscr{A} \subseteq \Omega$ is that of a σ-algebra.

[4] Probability assignments are assumed to satisfy $\mathbb{P}(\Omega) = 1$ (Ω can be interpreted as a tautological statement) and $\mathbb{P}(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$ for countably many mutually exclusive statements $A_1, A_2, \dots$ (i.e., at most one statement can be true at the same time). An interpretation of the first equation is that tautological statements are inevitable. The second equation states that the plausibility that one of some mutually exclusive statements is true equals the sum of the plausibilities of the single statements.

[5] Technically, only functions that are *measurable* are called random variables. This is, however, not really a restriction in practice. In fact, the author never encountered a non-measurable function in any application.

Using this notation, the probability that $\check{x}$ has a particular value $x \in X$ is then denoted by

$$\mathbb{P}(\check{x} = x) := \mathbb{P}(\check{x} \in \{x\}). \tag{2.2}$$

The random variable $\check{x}$ could here for example refer to a harmonic dependency structure of a tune to be analyzed. In that case, X would denote the set of all possible dependency structures of that tune and $x \in X$ would denote a particular structure.

In this chapter, random variables are always denoted using a check accent (e.g., $\check{x}$). As common in the machine learning literature, such accents are not written in the rest of this study. It should then be clear from the context when random variables $\check{x}$ and when concrete values x are meant.

Random variables which range over a finite or countable set are called *discrete*. For a discrete random variable $\check{x}$, there is a function $p\colon X \to \mathbb{R}_{\geq 0}$ such that $p(x) = \mathbb{P}(\check{x} = x)$ for all possible values $x \in X$. The function p is called the *probability mass function* of $\check{x}$ with respect to $\mathbb{P}$. A random variable that ranges over a subset of $\mathbb{R}^d$ for some dimension $d \in \mathbb{N}_{\leq 0}$ is called a *continuous random variable* if there exists a function $p\colon X \to \mathbb{R}_{\geq 0}$ such that for all coherent sets of possible values $Y \subseteq X$, $\mathbb{P}(\check{x} \in Y) = \int_{x \in Y} p(x)$.[6] The function p is called the *density function* of $\check{x}$ with respect to $\mathbb{P}$. For a unified simple terminology, probability mass functions are also referred to as density functions in the following.[7]

The function which maps a subset of possible values $Y \subseteq X$ to the probability $\mathbb{P}(\check{x} \in Y)$ is called the *distribution* of the random variable $\check{x}$. For a discrete random variable, $\mathbb{P}(\check{x} \in Y) = \sum_{x \in Y} p(x)$. Since this study assumes all random variables to be either discrete, continuous or a combination of both, each distribution has an associated density function.[8] By convention of the machine learning literature, the distribution of a random variable $\check{x}$ is denoted by $p(\check{x})$. Note that since the accent is omitted later in this study, the notation of a distribution clashes with the notation of the probability $p(x)$ of a value $x \in X$.

One of the most important properties of a random variable $\check{x}$ is its expected value $\mathbb{E}[\check{x}]$. It is defined by $\mathbb{E}[\check{x}] = \sum_{x \in X} p(x)\, x$ for discrete random variables and by $\mathbb{E}[\check{x}] = \int_{x \in X} p(x)\, x$ for continuous random variables.[9] In cases that benefit from explicitly denoting the underlying probability distribution, such distribution can be indicated as a subscript, like in $\mathbb{E}_{p(\check{x})}[\check{x}]$. This applies for instance to complex random variables constructed as transformations from simpler random variables. The transformation of a random variable $\check{x}$ by a function $g\colon X \to Z$ is denoted by $g(\check{x})$; it is defined by $(g(\check{x}))(\omega) = g(\check{x}(\omega))$ for a world state $\omega \in \Omega$. The expected value of the transformed random variable $g(\check{x})$ is thus denoted by $\mathbb{E}_{p(\check{x})}\left[g(\check{x})\right]$. For example, if $g(x) = x + y$ for $x, y \in X = \mathbb{N}$ and if random variables are not explicitly marked, like it is the

[6] A subset $Y \subseteq X$ is called *coherent* if it is a Borel set.

[7] This is common in mathematical probability theory. It is justified by the fact that probability mass functions are density functions with respect to the counting measure.

[8] This follows from the Radon-Nikodym theorem.

[9] The definition of the expected value of a discrete variable $\check{x}$ requires that the values over which $\check{x}$ ranges can be added and multiplied with a real number in a meaningful way.

case in later chapters, then $\mathbb{E}_{p(x)}\left[x+y\right]$ clarifies that the expectation is calculated with respect to the random variable x and the underlying probability measure $\mathbb{P}$.

2.4 The observation process: modeling learning as conditioning

The probability measure $\mathbb{P}$ implicitly includes all assumptions that the agent makes about the world. Model assumptions which are accepted throughout an experiment are usually treated that way. Other assumptions are incorporated using *conditional probability*; the probability measure constructed from $\mathbb{P}$ with an additionally assumed statement A is denoted by $\mathbb{P}(\cdot \mid A)$ and the corresponding distribution of a random variable $\check{x}$ by $p(\check{x} \mid A)$. The observation of data can for example be expressed as conditioning. One scenario considered in this study is that the agent reasons about the unknown grammar of Jazz harmony by observation of chord sequences. Both the grammar and the sequences are aspects of the world; their random variables are denoted by $\check{g}$ and $\check{\boldsymbol{w}} = (\check{w}^1, \dots, \check{w}^I)$, respectively, where $I \in \mathbb{N}$ denotes the number of sequences. The observed sequences are denoted by $\boldsymbol{w} = (w^1, \dots, w^I)$. The probability distribution $p(\check{g})$ represents the belief about the grammar before the observation of the sequences $\boldsymbol{w}$. It is called the *prior distribution.* The belief about the grammar after the observation is represented by the conditional distribution $p(\check{g} \mid \check{\boldsymbol{w}} = \boldsymbol{w})$, commonly abbreviated by $p(\check{g} \mid \boldsymbol{w})$. It is called the *posterior distribution.* The learning that takes place by the observation of $\boldsymbol{w}$ is thus represented by the transition from the prior distribution $p(\check{g})$ to the posterior distribution $p(\check{g} \mid \boldsymbol{w})$.

Conditional probability densities such as $p(g \mid \boldsymbol{w})$ for which $p(\boldsymbol{w}) \neq 0$ (i.e., the observation must have been possible) are calculated by

$$p(g \mid \boldsymbol{w}) = \frac{p(g, \boldsymbol{w})}{p(\boldsymbol{w})}, \tag{2.3}$$

where $p(g, \boldsymbol{w})$ denotes the density of the joint distribution of $\check{g}$ and $\check{\boldsymbol{w}}$. Analogously,

$$p(\boldsymbol{w} \mid g) = \frac{p(g, \boldsymbol{w})}{p(g)}. \tag{2.4}$$

By substitution of $p(\boldsymbol{w} \mid g)\, p(g)$ for $p(g, \boldsymbol{w})$ in Equation 2.3, *Bayes rule* is obtained,

$$p(g \mid \boldsymbol{w}) = \frac{p(\boldsymbol{w} \mid g)\, p(g)}{p(\boldsymbol{w})}. \tag{2.5}$$

In words, it states that the posterior belief is proportional to the prior belief multiplied by the *sample probability* $p(\boldsymbol{w} \mid g)$ — the probability that the observed data (here a set of chord sequences) was generated from the assumed grammar. Note that because Equation 2.3 states a property of densities, it is not true by definition but must be derived (Schervish, 2012).

Bayes rule is at the core of Bayesian statistics, as it formalizes learning from data by induction. Furthermore, it associates in a single equation the idea of statistical learning with structured domain knowledge that is encoded into the prior distribution. Critiques of Bayesian methods argue that the choice of the prior distribution brings undesired biases into statistical evaluations. Might that be true or not for other problems, the comparison of different prior distributions is a well-suited tool to study human's abstract knowledge about music, because both statistical learning and structured domain knowledge are necessary to explain the use and acquisition of human knowledge (Tenenbaum et al., 2006). As Griffiths et al. (2008) put it more generally:

> Most human inferences are guided by background knowledge, and cognitive models should formalize this knowledge and show how it can be used for induction. From this perspective, the prior distribution used by a Bayesian model is critical, since an appropriate prior can capture the background knowledge that humans bring to a given inductive problem.

The prior distribution $p(\check{g})$, the data-generating distribution $p(\check{\boldsymbol{w}} \mid g)$, and the observed data $\boldsymbol{w}$ are the three dependent factors mentioned Section 2.2. The distribution $p(\check{\boldsymbol{w}} \mid g)$ is an important part of the statistical model since it describes how the harmonic grammar assigns probabilities to chord sequences. For context-free grammars, there is a standard construction for this assignment, which is described in the next section.

2.5 Bayesian statistics and probabilistic generative modeling

So far, probability theory was presented along with its Bayesian interpretation. The application of the theory to empirical problems is the topic of statistics. Statistics thus acts as an interface between mathematics and the real world. More precisely, statistics can be considered the study of probabilistic models, their assumptions, and their interpretation. Therefore, the term *statistical model* is used to refer to a probabilistic model together with the assumptions it makes and the interpretation that is attributed to it.

But what is a probabilistic model? The aim of such a model is to relate the observed and the latent variables of the problem at hand. In Bayesian statistics, all observed and all latent variables are modeled as random variables and the uncertainty about unobserved variables is represented by their probability distributions. All relations between the variables are fully described by their joint distribution.

The models considered in this study are instances of a generic grammar-learning model. There are three kinds of random variables: one for the grammar of Jazz harmony, one for Jazz chord sequences, and one for derivation trees of such sequences. The chord sequences are observed and the grammar is latent. In a supervised learning setting, the derivation trees are observed, whereas they are latent for unsupervised learning. In this section, the random variables of the grammar, the derivation trees, and the sequences are denoted by $\check{g}$, $\check{\boldsymbol{t}} = (\check{t}^1, \ldots, \check{t}^I)$, and $\check{\boldsymbol{w}} = (\check{w}^1, \ldots, \check{w}^I)$, respectively. The random variable for the grammar includes information about the grammar's nonterminal representation and its rewrite rules. As described below, a grammar can be understood as a distribution over derivation trees. A distribution over grammars is therefore a distribution over distributions, much like the Dirichlet distribution is a distribution over categorical distributions (see Section 7.2). This will become clearer in Chapter 7 when distributions over grammars are discussed in detail. For now, it suffices to view the uncertainty about a grammar as a result of the uncertainty about the rules contained in the grammar as well as the uncertainty about how probable a rule is to be applied to generate a derivation tree.

The joint distribution of all random variables is synonymous to the probabilistic model, because all relations between the variables are captured in that distribution. Bayesian models are commonly specified in two parts:

1. a factorization of the joint distribution into a product of conditional distributions and
2. distribution assumptions for all of the conditional distributions.

The factorization describes which random variables depend on which other variables and the conditional distributions define how random variables relate to the variables they directly depend on. In the example of grammars, derivation trees, and chord sequences, the density of the joint distribution is factorized by

$$p(g, \boldsymbol{t}, \boldsymbol{w}) = p(g) \prod_{i=1}^{I} p(t^i \mid g)\ p(w^i \mid t^i). \tag{2.6}$$

The probability mass function $p(w^i \mid t^i)$ is simple; it returns 1 if w^i is the leaf sequence of the tree t^i and 0 otherwise. The density functions $p(g)$ and $p(t^i \mid g)$ are more complicated and defined below. This probability model is standard for context-free grammars (Kurihara and Sato, 2004, 2006; Johnson et al., 2007b). It is furthermore an instance of a class that is called *probabilistic generative models* (Bishop, 2006; MacKay, 2003). A model is defined as being part of that class if the dependency structure represented by the factorization of the joint distribution is acyclic. The dependency structure of the grammar model is acyclic as shown in Figure 2.1. Given any grammar, all derivations trees are independent and a chord sequence depends only on its derivation tree. If otherwise, the grammar and the derivation trees are not known, then all chord sequences depend on each other.

Generative models are called generative, because they have an interpretation as a generative process in which the random variables are generated one at a time. The generation can be done in any order that respects the dependency structure of the model, that is a variable can be generated only if all variables it depends on are already generated. In the grammar example, one possible order is to first generate the grammar, then all derivation trees, and then all chord sequences.

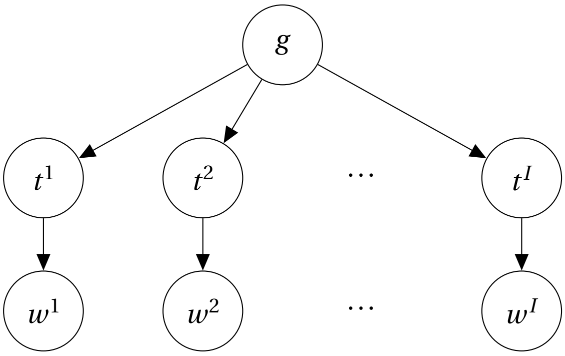


Figure 2.1 – Dependency structure of the generative model of a grammar g, derivation trees $t^1,\ldots,t^I$, and chord sequences $w^1,\ldots,w^I$.

To define the distribution $p(\check{t} \mid g)$ of a derivation tree t given a grammar g, the definition of context-free grammars is extended to probabilistic context-free grammars. Such probabilistic grammars also have an interpretation as a generation process of derivation trees, formalizing the more intuitively presented generation procedure of context-free grammars in the last chapter. For the random generation, each nonterminal is equipped with a distribution of rules which have that nonterminal as their left-hand side (e.g., that rewrite the nonterminal into a sequence of terminals and nonterminals). The generation procedure starts by generating a rule to rewrite the start symbol into a sequence of terminals and nonterminals. Afterwards, it iteratively generates rules to rewrite the leftmost nonterminal of the sequence until the sequence only consists of terminals. Then, the process halts and the derivation tree including the terminal sequence is returned.

Probabilistic context-free grammar. A *Probabilistic Context-Free Grammar* (PCFG) is a context-free grammar (T, N, Start, R) in which each rule $(A \longrightarrow \boldsymbol{\alpha}) \in R$ is associated with a positive real number $p(A \longrightarrow \boldsymbol{\alpha})$ such that

$$\sum_{\boldsymbol{\alpha} \in (T \uplus N)^*} p(A \longrightarrow \boldsymbol{\alpha}) = 1 \tag{2.7}$$

for all nonterminals $A \in N$, where $\boldsymbol{\alpha}$ denotes a sequence of terminals and nonterminals. A number $p(A \longrightarrow \boldsymbol{\alpha})$ is then interpreted as the probability that the rule $A \longrightarrow \boldsymbol{\alpha}$ is sampled to rewrite the nonterminal A. The probability of a derivation tree is defined as the product of the probabilities of the rules that constitute that tree. The probability of a sequence of terminal symbols is defined as the sum of the probabilities of all its derivations.

Probabilistic grammars are described more elaborately in Section 5.4. For now, it suffices to note that a PCFG defines a distribution $p(\check{t} \mid g)$ over derivation trees t where the rule probabilities are included in the representation of the PCFG g. All information about a PCFG is contained in the function $(A \longrightarrow \boldsymbol{\alpha}) \mapsto p(A \longrightarrow \boldsymbol{\alpha})$ that maps rules to their probabilities, where rules which are not contained in the grammar are assigned to zero. The distribution $p(\check{g})$ that models the prior knowledge of the agent can thus be understood as a distribution over such mappings. The concrete definition of prior distributions will be subject to experimentation in the computational learning experiments presented in the third part of the thesis. To allow for a wide range of distributions $p(\breve{g})$, this study proposes probabilistic abstract context-free grammars in Chapter 5. A general class of prior distributions which will be used for grammar learning is then described in Chapter 7.

2.6 The computational level of cognition

Bayesian models provide the opportunity to rigorously study cognitive representations without the need to assume any details about human processing mechanisms. They are therefore well-suited for this study which is concerned with the learnability of cognitive representations. Even the learning process is modeled without specific assumptions about learning mechanisms; it is characterized by the relation between two distributions, the prior and the posterior. In the grammar example, the prior distribution $p(\breve{g})$ represents the agent's (e.g., an ideal learner's) belief about the grammar before learning, and the posterior $p(\breve{g} \mid \boldsymbol{w})$ represents the belief after the learning process. Learning is thus modeled as the transition from the prior to the posterior.

The distinction of *what* the mind computes from *how* it does it was first proposed by Marr (1982) for the study of vision. Marr's proposal aimed at a better understanding of high-level cognitive processes and contributed to an improvement of theory development in cognitive sciences (Anderson, 1990). Concretely, Marr proposed to distinguish three levels of cognition:

Computational level: What mathematical function is computed?

Algorithmic level: What algorithm is used to compute the function?

Implementational level: How is the algorithm physically implemented?

The implementational level is different for biological and artificial systems. In biological systems such as mammals it is realized by neural structures, and in artificial computing systems by electronic transistors. The computational and the algorithmic level are, however, similar for both systems.

An analogy from computer science can aid the understanding of the three levels. The computational level describes the mathematical function that is computed, for example the sum of a list of integers. That level does not concern how such a sum might be computed, the only thing that matters is that a list of integers is mapped to the sum of its elements. A description on the computational level is thus similar to a specification of a computer program. A statement of a mathematician that she proved some function exists without knowing an explicit construction of it could also be classified into the computational level. The algorithmic level can be understood as describing algorithms as programs in any assembly language (e.g., in primitive machine instructions). Commonly, such programs are written a high-level language such as python or haskell and the high-level language is then interpreted in or compiled to an assembly language. In the example of the sum of a list, a high-level program would take a list as input and return the sum of it. Valid programs for this simple case include the summation from left to right, resulting in the term $(((((1+2)+3)+4)+5)+6)+7$ for the input list $[1,2,3,4,5,6,7]$, or a program which recursively splits lists into half and sums the smaller lists, resulting in $((1+2)+(3+4))+((5+6)+7)$. Since the second program can be easily parallelized, it can run much quicker than the first program when multiple processor cores are available. This illustrates another difference between the computational and the algorithmic level; while the former does not have a notion of time, the latter can compare algorithms in terms of their runtime complexity. While the algorithmic level can be understood in terms of programs written as machine instructions, the implementational level can be understood as the implementation of the machine instructions in physical computer hardware.

Coming back to the generic grammar model used in this study, the mathematical function that is computed maps the prior distribution $p(\breve{g})$ and the observed chord sequences $\boldsymbol{w}$ to the posterior distribution $p(\breve{g} \mid \boldsymbol{w})$. The probabilistic model characterized by the joint distribution of the grammar, the derivations trees, and the sequences constitutes the specification at the computational level. Bayes' rule then seems to correspond to the algorithmic level, because it is a formula that describes how to compute the posterior:

$$p(g \mid \boldsymbol{w}) = \frac{\sum_{\boldsymbol{t}} p(g, \boldsymbol{t}, \boldsymbol{w})}{p(\boldsymbol{w})} = \frac{\sum_{\boldsymbol{t}} p(g, \boldsymbol{t}, \boldsymbol{w})}{\int_g \sum_{\boldsymbol{t}} p(g, \boldsymbol{t}, \boldsymbol{w})} \tag{2.8}$$

This is in practice, however, not the case, because the calculation of the integral in the denominator is intractable. Therefore, algorithms which provide approximations of the posterior such as Markov chain Monte Carlo (MCMC) or variational Bayesian inference methods correspond to the algorithmic level. There is even some evidence for the "Bayesian coding hypothesis" which states that brains indeed represent uncertainty by approximation of probability distributions (Knill and Pouget, 2004; Brighton and Gigerenzer, 2008; Friston, 2009, 2010; Clark, 2013b). However, "[p]erhaps the greatest open question about Bayesian network and Bayesian learning models is how they might be implemented in the brain" (Gopnik and Tenenbaum, 2007).

Finally, note that Marr's computational level is loosely related to Chomsky's concept of competence. In *Aspects of the Theory of Syntax*, Chomsky (1965) proposed the distinction of *competence* from *performance* for natural language syntax. The former describes the ideal language system possessed by native speakers. It enables for example the distinction between grammatical and ungrammatical sentences. The latter, performance, describes how the language system is used in communication. In Chomsky's view, many sentences of spoken language which seem ungrammatical can be thus understood as grammatical sentences perturbed by speech errors. As mentioned above, there is some similarity between competence and the computational level, for example the abstraction from time, but as Anderson (1990) wrote:

> The relationship between competence and performance is really not the same as the relationship between Marr's level of computational theory and his lower levels [the algorithmic and the implementational level]. In Marr's case, the lower levels achieve the goals of the computational level. Chomsky's competence level is a theory based on a certain subset of data that is thought to be a direct and reliable reflection of the person's linguistic knowledge. For instance, judgments of whether a sentence is grammatically well formed provide key data for a theory of competence, but time to understand a sentence is thought to be less stable and is consigned to a theory of performance. Performance is somehow constrained to reflect the competence, but it reflects other factors as well. Unlike Marr's case, performance is not just a matter of implementing the goals of competence. Indeed, unlike Marr's computational-level, Chomsky's competence is not concerned with the goals of the system. A computational-level theory of language would have to be concerned with the functionality of language — a concern that Chomsky explicitly rejected.

In the context of this study, competence is the ideal system of harmonic structure that corresponds to the unknown grammar the agent reasons about. It is thus used as a representation on the computational level. For example in *Jazz Improvisation: A Theory at the Computational Level*, Johnson-Laird (1991) uses derivation trees to describe the structure of Jazz chord sequences at the computational level. In contrast, Ogura et al. (2020) apply Earley's parsing algorithm (Earley, 1970) to suggest how the creation of a derivation tree could be realized at the algorithmic level.

3 Related approaches to musical structure

The present study is a continuation of previous research that applied formal grammar models to analyse musical structure. The approach used in this study is largely based on the grammar models proposed by Rohrmeier (2011, 2020a) and Rohrmeier and Neuwirth (2015) as introduced in the first Chapter 1. Those models were earlier also implemented and applied to model harmonic similarity (De Haas et al., 2009) or to improve automatic chord recognition (De Haas, 2012; De Haas et al., 2012). This chapter gives an overview over related formalisms and models of musical structure. The three most related previous approaches are discussed in greater detail: 1) grammar models that directly formalize aspects of Schenkerian theory, 2) models based on the Generative Theory of Tonal Music (GTTM), and 3) models directly related to the Jazz-harmony grammar proposed by Granroth-Wilding and Steedman (2014).

The term *harmonic syntax* is used both by traditional music theory and research on formal models of harmony. Aldwell and Schachter (2003) define the term harmonic syntax based on the order how chords appear in a sequence:

> In studying music, we can use the term harmonic syntax to refer to the arrangement of chords to form progressions; the order of chords within these progressions is at least as important as the order of words in language.

This definition is very general since it does not mention the kinds of structural dependencies between chords that constitute the syntactic structure. In contrast to linguistics where syntax commonly refers to a hierarchical structure between the words of a sentence, approaches which model only local relations between chords might also be considered studies of harmonic syntax. Rohrmeier and Pearce (2018) and Pearce and Rohrmeier (2018) present an overview over computational models of harmonic syntax of different complexity, including local, linear, and hierarchical models.

Following up on the formalization of generative grammar by Chomsky (1965), scholars started to apply similar grammar models to music in the 1960s. Roads and Wieneke (1979) and Sundberg and Lindblom (1991) review early approaches to hierarchical structure in music that use formal grammars. The first formal grammar for Western tonal harmony known to the author was formulated by Winograd (1968). Later, Sundberg and Lindblom (1976) complement that model by sketching a formal grammar for melody to argue that both melodies and natural language syntax exhibit hierarchical constituent structure. Baroni and colleagues discussed the concept of musical grammar in greater detail in relation to natural language and cognitive science, and propose grammar rules for simple melodies of Baroque music (Baroni and Jacoboni, 1975; Baroni et al., 1983; Baroni and Jacoboni, 1983; Baroni et al., 1992; Baroni, 1999).

Many of recent grammar models are probabilistic (McCormack, 1996; Gilbert and Conklin, 2007; Abdallah and Gold, 2014; Abdallah et al., 2016; Tsushima et al., 2020). For example Quick (2010, 2014, 2016) and Quick and Hudak (2013a,b) use probabilistic grammars for automatic music generation and composition. Other approaches propose alternative grammar forms or algorithms for music analysis (Tidhar, 2005; Tojo et al., 2006; Sidorov et al., 2014).

A more theoretical question concerns the right level of comparison between musical and linguistic structure (Asano and Boeckx, 2015); there is no general agreement about the nature of the relation between music and language (Rebuschat et al., 2011; Arbib, 2013). Katz and Pesetsky (2011) hypothesize in their *Identity Thesis for Language and Music* that musical and linguistic structure are as similar as they can be. This study takes the rather different standpoint that both language and music are similar, because they are shaped by domain-general cognitive principles. That is, the capacity for musical skills such as improvisation and composition is assumed to be not directly related to language skills such as speaking and writing, but to general capabilities that also empower humans to drive and navigate a car, play sports, or organize a birthday party. A domain-general capacity needed for all those tasks is for example the ability to hierarchically decompose complex structures or tasks into simpler ones.

3.1 Schenkerian theory and formalization

Schenkerian theory (also known as Schenkerian analysis) was the first rigorous theory of hierarchical structure in music. It originated from Heinrich Schenker (1935) and can nowadays be considered a predecessor of modern computational models for musical structure. The theory was further developed by scholars such as Salzer (1952), Salzer and Schachter (1989), and Cadwallader and Gagné (2007); see Forte (1959) for a brief introduction. Schenkerian theory comprises a complex system of interacting rules for structural reduction of pieces from Western classical music. The reduction rules can be understood as inverse generation rules. The rule system is meant to be used by human experts of the theory to depict the logical coherence of a piece by stepwise reduction to an *Ursatz* (fundamental structure) which is assumed to be existent on the deepest structural level of all musical styles Schenkerian theory is applied to. As in a context-free grammar, similar transformational principles apply recursively at all levels of a Schenkerian reduction (e.g., an analysis of a piece), and the *Ursatz* (fundamental structure) is analogous to the start symbol of a context-free grammar.

Human expertise is necessary to interpret both the theory and the musical piece to be analyzed to address their ambiguities. Schenkerian theory is thus itself commonly considered an artistic activity with the goal of finding new *hearings* of a piece (i.e., new ways to hear a piece). Schenkerian theory relates to the study of music perception, because it considers the question of how a piece is perceivable. However, Temperley (2011) for example argues that it is not straight-forward to evaluate Schenkerian theory as a theory of music perception. Cross (1998) and Neuwirth and Rohrmeier (2016) acknowledge the methodological differences between music analysis and music cognition, and suggest that they should complement each other.

The generative principles of Schenkerian theory can be generalized for styles other than Western classical music. Recent research applied them, for example, to Jazz (Martin, 1996, 2011a,b; Larson, 1998, 2002, 2009; Givan, 2010; Heyer, 2012) and Rock music (Everett, 2004). Applications to other styles remain largely unexplored (Stock, 1993; Clarke, 2017). Generalizations of Schenkerian analysis commonly inherit the recursive nature of the theory. In such generalizations, the conception of the Ursatz is, in contrast, a topic of the scientific discourse: The specific form of the Ursatz proposed by Schenker consists of a descending melody from scale degree three or five to the root of the key, accompanied by a harmonic I-V-I movement. That form does not apply to all styles of Western tonal music or even all music in general.

In their article *A LISP-Based System for the Study of Schenkerian Analysis*, Frankel et al. (1976) reported “the first attempts at modeling musical perception on a digital computer using the methodology of Heinrich Schenker’s theory of music.” They further developed their ideas in the next years (Frankel et al., 1978; Smoliar, 1979) using tree-structured data representations and transformational grammar (Chomsky, 1965). Their main findings were that computational modeling helps to clarify the ideas of Schenkerian analysis and can help to find parts where the theory is not explicit in a formal sense. Mavromatis and Brown (2004) find similar results much later by aiming to implement Schenkerian theory in the programming language Prolog using the formalism of definite clause grammars. Keiler (1978) proposes a simple phrase-structure grammar for harmony of Western classical music, inspired by Schenkerian analysis. Marsden (2001, 2005) uses directed acyclic graphs for the formalization of Schenkerian theory to represent musical relations that are not tree-structured. Marsden (2007, 2010) and Kirlin and Utgoff (2008) present implementations towards automatic Schenkerian analysis and conclude that a scoring function is needed to distinguish plausible from implausible analyses.

Yust (2006, 2015) propose to use what are called Maximal OuterPlanar graphs (MOPs) to formalize reductions of Schenkerian theory. MOPs are equivalent to derivations of a context-free grammar whose nonterminals are intervals between notes. The MOP model was extended by Kirlin and Jensen (2011, 2015), Kirlin (2014), and Kirlin and Thomas (2015) using probabilistic context-free grammars. Yust’s approach was also expanded by Finkensiep et al. (2019) to define a formal grammar for North-Indian Raga melodies.

3.2 The generative theory of tonal music

The Generative Theory of Tonal Music (GTTM) focusses on the musical idiom of Western Classical music also known as the common practice period (Lerdahl and Jackendoff, 1983). It was later also applied to popular music such as Beatles' songs (e.g., Jackendoff and Lerdahl, 2006). A main goal and the core achievement of the GTTM was to link hierarchical music analysis that originated from Schenker with music cognition (Giblin, 2008). This was accomplished by reinterpreting parts of Schenkerian theory as a recursive formalism to describe cognitive representations of experienced listeners. Much following research was based on this connection of music theory and cognitive science as well as on the relation between music and language stressed by the GTTM (Bigand et al., 2009) as described below.

Lerdahl (2009) gives a review of the "Genesis and Architecture of the GTTM project". The origin of the GTTM was Noam Chomsky's reformulation of linguistic theory as the formal study of the human capacity for language (Chomsky, 1965). While Bernstein (1976) advocated for a literal transfer of linguistic concepts to musical concepts, Lerdahl and Jackendoff (1983) were more interested in the general spirit of Chomsky's research program including the distinction between competence and performance as well as the idea of a small set of formal rules that can be used to generate infinitely many sentences. The rules system proposed by the GTTM is, however, formally different from rule systems of generative grammar used in linguistics and also by this study. Instead of generative rules, the GTTM uses preference rules to contrast the gradient nature of music phenomena with the categorial grammatical-or-not-grammatical distinction of phrase-structure rules in the 1970ths. The way that the GTTM uses preference rules was criticized as being not sufficiently quantified (Peel and Slawson, 1984) and considered "a quasi-formal description of the different musical structures that underlie the perception of Western music" (Bigand et al., 2009). This study, as well as other modern approaches, acknowledges the ambiguity of music using probability theory.

The GTTM models components of musical structure such as meter and pitch independently and combines the analyses of the components to a overall structural description of a musical piece. One important contribution of the GTTM is the disentanglement of grouping and meter, two structures that were commonly confused before (Cone, 1968; Cooper and Meyer, 1960). That distinction is now perceptually validated, widely accepted, and assumed by many studies (Deliege, 1987; McAdams, 1989; Palmer and Krumhansl, 1990; Large and Palmer, 2002; Frankland and Cohen, 2004; Yust, 2018). Grouping and meter lead to a time-span reduction by the application of stability conditions. The time-span reduction represents aspects of rhythm as a hierarchical time structure. The GTTM generally uses the term reduction synonymously to hierarchy. Additional to meter and grouping, pitch is considered as a third dimension of music. The application of well-formedness rules and preference rules to the pitch-structure of a piece leads to a prolongational reduction. The prolongational reduction trees proposed in the GTTM are a hierarchical representation of a piece's tonal tension and relaxation. They are loosely related to the prolongational structure in Schenkerian theory. The time-span reduction and the prolongational reduction thus represent complementary aspects of a piece. However, the prolongational reduction is partly derived from the rhythmic stabilities represented in the time-span reduction through an interaction principle.

The GTTM interprets prolongational branchings as patterns of tonal tension and relaxation. Right branches and left branches represent an increase and decrease in tension, respectively. Branches are of one of three types: *strong prolongation* (exact repetition), *weak prolongation* (small increase or decrease in tension), and *progression* (greater increase or decrease in tension). Following research studied the tension-relaxation system more extensively (Bigand et al., 1996; Krumhansl, 1996; Schellenberg, 1996; Lerdahl, 1996; Lerdahl and Krumhansl, 2007). Lerdahl (2009) writes that

> [...] the prolongational component as set forth in GTTM is not quantifiable. Its branching types describe degrees of tension and relaxation only qualitatively. And GTTM gives merely a verbal sketch of the stability conditions that underlie both the time-span and prolongational components.

The shortcomings of the GTTM such as the qualitative nature of the prolongational component lead to the development of *Tonal Pitch Space* (TPS; Lerdahl, 1988, 2001). TPS quantifies stability conditions based empirical data of the tonal hierarchy (Krumhansl, 1983, 1990). For example, Yamamoto et al. (2020) very recently analyzed Jazz chord sequences as paths in TPS.

Hamanaka, Hirata, and Tojo studied computational applications of the GTTM extensively over many years (Hamanaka et al., 2005, 2006, 2007a,b, 2014, 2015, 2016b,a, 2020; Hamanaka and Tojo, 2009; Groves, 2016). The implementation was partly made difficult by the fact that the GTTM was originally not meant to perform automatic analyses but to provide a quasi-formal model for music analyses by human experts.

This study goes a different path than proposed by the GTTM and followed by Hamanaka, Hirata, and Tojo. On one hand, the modeling goals, close relation to music cognition, and abstract ideas such as the preliminary separation of musical dimensions like rhythm and harmony are similar, on the other hand, this study approaches the music from a different angle. Instead of weakening the formality of the rule system to address the ambiguous nature of music as done in the GTTM, complex probabilistic models over simple rule systems of rigorous formality are used. They are then investigated in order to see how much musical structure can be captured by such simplicity.

3.3 Combinatory categorial grammars for harmonic structure

From all of the approaches to musical structure discussed in this chapter, the one by Mark Steedman and Mark Granroth-Wilding might be most similar to the approach of this study, both because of formal and philosophical commonalities. The differences are found in details and notation. The grammar by Steedman and Granroth-Wilding evolved in three steps and was influenced by earlier and related research in computational modeling of music cognition —

most notably by Christopher Longuet-Higgins, a pioneer of music cognition in particular and cognitive science in general (Longuet-Higgins, 1976; Steedman, 1977; Longuet-Higgins, 1979; Longuet-Higgins and Lee, 1984; Longuet-Higgins and Lisle, 1989). The first version of the grammar was a proposal for the harmonic structure of the 12-bar Blues which evolved into a more general grammar for Jazz chord sequences that constitutes the third version. Simpler context-free grammars for harmonic analysis of Jazz chord sequences influenced by the first version of the grammar were for example implemented by Pachet (1997), Chemillier (2004), and Katz (2017).

The first version of the model was a context-sensitive grammar for the 12-bar Jazz Blues (Steedman, 1984). The grammar was formulated using scale degrees, and thus abstracts form the root of a tunes's key. The formalism acknowledges that the grammar rules can be applied at various metrical levels, while rules used in the examples of the paper mostly split chord durations equally. The set of rules includes a 6-ary start rule, rules for prolongation, subdominant departure, dominant preparation, and tritone substitution. Which chords are allowed to substitute other chords is regarded with respect to musicians' intuition (Steedman, 1984):

> [...] where a rule of the grammar says that one sequence of chords may replace another, musicians should agree that the substitution is a possible expression of such aspects of the musical meaning as the underlying cadential sequence.

In the terminology introduced in Chapter 1, "possible expression of such aspects of the musical meaning as the underlying cadential sequence" means that the replaced (e.g., substituted) chord sequence must correspond to the same harmonic dependency structure.

The second version of the grammar is a reformulation of the first version as a Combinatory Categorial Grammar (CCG) that aims to overcome the idea of chord substitution (Steedman, 1996). CCG is a "radically lexicalized theory of grammar" which separates language-specific syntactic information from language-independent rules (Steedman, 2019; Steedman and Baldridge, 2006). The language-specific information is stored in a lexicon by assignment of categorial types to words. Such categorial types include information about part of speech, directionality, agreement, and semantic interpretation. CCG originated from the study of natural language and comprises a transparent interface between syntax and semantic representation. In contrast to Chomsky (1965) who considers syntax as an ideal system of structure, Steedman (2000) considers it as the process by which semantic interpretation is derived in a compositional way. In Steedman's view, the semantic representation is understood as the goal of the syntactic process. The application of CCG to music thus needs to define what is regarded as the semantics of harmony. Steedman (1996) chooses paths in the *Tonnetz* (Euler, 1739; Cohn, 1997; Gollin, 2006; Moss, 2019) as the semantic representation of chord sequences, as proposed by Longuet-Higgins (1962, 1979) and Longuet-Higgins and Lisle (1989). This semantics postulates that musically coherent chord sequences are chord sequences whose roots progress in small steps on the Tonnetz. The formalism therefore focuses on cadential chord progressions. For example, a simple chord sequence II V I progresses in minimal steps on the Tonnetz. More advanced examples which are generated using substitutions are analyzed as non-substituted chords, for instance the tritone-substituted progression II bII I is analyzed as the same root progression as II V I.

The third and last version of the model to date is an extension of the second version to more general Jazz chord sequences (Granroth-Wilding, 2013; Granroth-Wilding and Steedman, 2014). The semantic representation is adopted from the second version and now in the third version fully formalized: "The harmonic interpretation of a piece is the path through the tonal space traced by the roots of the chords" (Granroth-Wilding and Steedman, 2014). The third version of the grammar increases the lexicon of musical rules and makes use of additional domain-general combination principles such as coordination. The coordination rule is used to combine two unresolved cadences into a single unresolved cadence. It does therefore not distinguish double-preparations of tonics from prolonged dominant preparations. For example, a chord sequence V V I could be grouped either ((V V) I) or (V (V I)), and both are understood as coordination of the fifth scale degree.

The harmonic dependency structures considered in this study are similar to the semantic representations used by Granroth-Wilding and Steedman (2014). For example, neither approach uses the grammar rules to restrict the set of possible modulations, regularities are instead captured by the statistical models. One difference is that the semantic representations focus on cadential chord progressions and do not model high-level organization of Jazz tunes such as hierarchical phrase structure and form: "[...] a piece of music is analysed as a sequence of expectation-resolution structures and no structure is analysed between these fragments" (Granroth-Wilding and Steedman, 2014). Large-scale harmonic dependencies such as dominant-tonic relations between B and A sections in an AABA from (e.g., in *Take the "A" train*) or between B and C in an ABAC form (e.g., in *All of me*) are thus not modeled explicitly. Other differences are that this study jointly models harmony and rhythm and also tackles the task of grammar induction.

3.4 Linear and local models of musical sequences

Local models of sequential structure such as n-gram models are different from context-free grammars in that they cannot represent dependencies over long distances. Linear models such as Hidden-Markov-Models (HMMs) and artificial recurrent neural networks such as Long Short-Term Memory (LSTM) are additionally able to represent non-local dependencies in principle. In practice however, there is a limit to which extent such non-local dependencies are learnable. Probabilistic context-free grammars can suffer from the opposite problem that long-range dependencies are found which do not describe the music well, as shown in the supervised computational experiments in Chapter 9.

The focus of this study are hierarchical analyses of chord sequences. Since these structures are not straight-forward to study with local and linear models, such models are rather loosely related to this study. A more natural application of local or linear models is the task of predicting the next note with which a sequence of notes most likely progresses (Pearce and Wiggins, 2006). The following paragraphs give a brief overview of local and linear models for harmony used in previous research. See for example Rohrmeier and Pearce (2018) and Pearce and Rohrmeier (2018) for a more detailed review and Rohrmeier and Graepel (2012) for an empirical study comparing different model classes.

The simplest n-gram models are unigram models ($n = 1$). They are also called bag-of-notes models (or bag-of-chords), because they completely abstract from temporal order in which musical events occur. Recently, Yust (2019) studies historical developments in musical style using a unigram model of pitch classes for beginnings, endings, and whole pieces. Temperley (2018) and Temperley and de Clercq (2013) used unigram models to analyse harmony and melody in Rock and Pop music.

Bigram models (i.e., 2-gram models), also called Markov models or Markov chains, can be used to describe relations between two successive musical events. They are for example used to describe statistical regularities of chord transitions to study Jazz improvisation (Pfleiderer et al., 2017; Frieler, 2014, 2019, 2020), harmonic schemata in Jazz standards (Shanahan and Broze, 2012), harmony in Rock music (De Clercq and Temperley, 2011), harmony of Western popular music (Shaffer et al., 2020), harmony in Beethoven's string quartets (Moss et al., 2019; Moss, 2019), harmony in Bach's chorals (Rohrmeier and Cross, 2008), and the stylistic evolution of Western classical music (Rodriguez Zivic et al., 2013). A music-theoretic predecessor to bigram models is the table of usual root progressions in Bach's chorals proposed proposed by Piston (1948), in which he presented his qualitative estimation which chord progression occurs how often. Tymoczko (2006, 2011) also uses local models to study chord progressions as linear motions in multidimensional, non-Euclidean geometric spaces.

n-gram models are commonly used to describe a single musical dimension such as melody, harmony, or rhythm. Multiple viewpoint models extend n-gram models by taking more than one musical dimension in account (Conklin and Cleary, 1988; Conklin and Witten, 1995; Pearce, 2005a; Pearce et al., 2005; Whorley et al., 2013; Cherla et al., 2013). The Information Dynamics Of Music (IDyOM) model extends multiple-viewpoint models further to model both long-term memory and online learning (Pearce, 2005b, 2018) Skipgram models are another generalization of n-gram models orthogonal the the multiple-viewpoint idea. They allow for discharging some elements of a sequence such as errors or ornamentation to consider non-contiguous constituents. Skipgram models were recently used to study voice-leading schemata (Sears et al., 2017; Finkensiep et al., 2018; Sears and Widmer, 2020)

Hidden-Markov models are linear models that use a latent state to predict continuations of sequences. They were for example used for chord segmentation (Sheh and Ellis, 2003), chord recognition (Lee and Slaney, 2006; Khadkevich and Omologo, 2009), and key estimation (Peeters, 2006; Lee and Slaney, 2008). More recent models use artificial recurrent neural networks and, specifically, long short-term memory (LSTM) models for chord recognition, music generation, and transcription tasks (Boulanger-Lewandowski et al., 2012, 2013; Sigtia et al., 2015; Zhou and Lerch, 2015; Hadjeres et al., 2017; Korzeniowski and Widmer, 2016, 2018; Korzeniowski et al., 2018). However, a recent study by Wu and Yang (2020) shows that also modern attention-based artificial neural networks fall short on modeling idiomatic high-level structures of Jazz tunes, providing another motivation for this study.

Methods and Data Part II

4 The Jazz harmony treebank[1]

A critical resource for building and evaluating grammatical models of harmony is a ground-truth database of syntax trees that encode hierarchical analyses of chord sequences. This chapter introduces the Jazz Harmony Treebank (JHT), a dataset of hierarchical analyses of complete Jazz standards. The analyses were created and checked by experts, based on lead sheets from the open iRealPro collection[2]. We report on the creation of the treebank, elaborate on the musical interpretation of the syntax trees, and explain the decisions that were made to meet the challenges of the annotation procedure. The JHT is publicly available in JavaScript Object Notation (JSON), a human-understandable and machine-readable format for structured data.[3] Additionally, statistical properties of the corpus are summarized and a simple open-source web application for the graphical creation and editing of trees is presented which was developed during the creation of the dataset.

The major challenge of the JHT's creation process lies in the many individual decisions analysts have to take to address the ambiguity of music. That is, some chord sequences can be heard in multiple ways, and the analyst has to decide which way describes the harmonic dependency structure of the tune best. Importantly, the goal is not to create uniform syntax trees of Jazz chord sequences, but to describe individual and subjective listening experiences in an unambiguous formal representation. Harmonic relations in sufficiently long chord sequences can be perceived in several ways, without one interpretation being clearly preferable. Therefore, the syntax trees of the JHT are best understood as proposals with a clear interpretation. The trees provide a basis for further analytical discussions, for education, and for training and evaluation of the grammar models in the computational experiments presented later in the

[1] This chapter is based on a peer-reviewed article which was accepted for publication: Harasim, D., Finkensiep, C., Ericson, P., O'Donnell, T. J., and Rohrmeier, M. (2020). The Jazz Harmony Treebank. *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR)*. Author contributions: DH created most of the initial tree analyses and wrote the first draft of the paper. The remaining tree analyses were initially created by a student assistant. DH, CF, & PE revised the initial tree analyses. DH and CF designed and implemented the tree creation app. DH and PE created the plots and calculated the summary statistics. All authors contributed to the conceptualization of the research and the writing of the text.

[2] https://irealpro.com/

[3] The treebank is available at https://github.com/DCMLab/JazzHarmonyTreebank.

third part of this thesis.

The scope of the treebank is limited to tonal Jazz, including Swing, Bossa Nova, Jazz Blues, Bebop, Cool Jazz, and Hard Bop, and excluding parts of traditional Blues, Modal Jazz, Free Jazz, and Modern Jazz. Tunes such as *Groovin' high* and *Out of nowhere* whose harmonic structure requires even more expressive representations than trees are excluded.[4] The general idea of harmonic syntax is, however, also applicable to other musical styles such as Western classical music (Rohrmeier, 2011; Rohrmeier and Neuwirth, 2015).

4.1 Related datasets

Treebanks are of particular importance for the study of hierarchical models and their applications. For example in linguistics, they have been and remain instrumental for many natural language processing tasks. The well-known Penn Treebank(Marcus et al., 1993), first published in the early nineties, is an instructive example since it has been used as an object of study in and of itself (Gaizauskas, 1995), as a basis for publishing additional treebanks with different paradigms (Hockenmaier and Steedman, 2007) and for different languages (Maamouri et al., 2004), and–most prominently–as a dataset for training and evaluating machine-learning methods (Katz-Brown et al., 2011; Sarkar, 2001; Melis et al., 2017).

Many existing collections of symbolic data about chord sequences concentrate on providing chord labels for harmonic entities. Harte et al. (2005) proposed a structured representation of chord symbols that they applied to label the audio data of the complete Beatles collection with time alignment. Burgoyne et al. (2011) provide an extended dataset of time-aligned chord symbols in a similar format for songs of popular music. These two datasets were primarily created to study automatic chord transcription from audio. Neuwirth et al. (2018) and Moss et al. (2019) take a more music-theoretically motivated approach by proposing a chord-symbol representation for Western classical music and apply it to scale degree analyses of Beethoven's string quartets. Chen and Su (2018) and Devaney et al. (2015) similarly label excerpts of sonatas, madrigals, chorals, preludes, and songs from common-practice tonality. Micchi et al. Micchi et al. (2020) combine existing Roman numeral analyses into a meta-dataset.

The datasets just mentioned use chord labels to analyze music given as audio data or in a symbolic representation. Since this study analyzes the relations between the chords of such sequences, it is located at a higher level of abstraction. Only a few datasets of hierarchical analyses of sequential musical data are available in divergent formats (Rizo and Marsden, 2016). Hamanaka et al. (2014) and Kirlin (2014) created two datasets of tree analyses of melodies of Western Classical Music based on the *Generative Theory of Tonal Music* (GTTM; Lerdahl and Jackendoff, 1983). Gotham and Ireland (2019) study musical form by the creation of datasets in a hierarchical representation. Moss et al. (2020) study Brazilian Choro using a dataset with hierarchical form encoding. Granroth-Wilding and Steedman (2014) provide

[4] *Groovin' high* exhibits crossing harmonic dependencies between a tonic prolongation from m1 to m5 and a dominant preparation from m4 to m7. *Out of nowhere* has a similar structure.

a dataset of 76 sub-sequences of Jazz standards with partial harmonic grouping labels. In contrast to previous research that analyzed snippets of musical pieces, the JHT consists of 150 full chord sequences of Jazz standards with complete harmonic syntax trees.

4.2 Complete constituents and open constituents

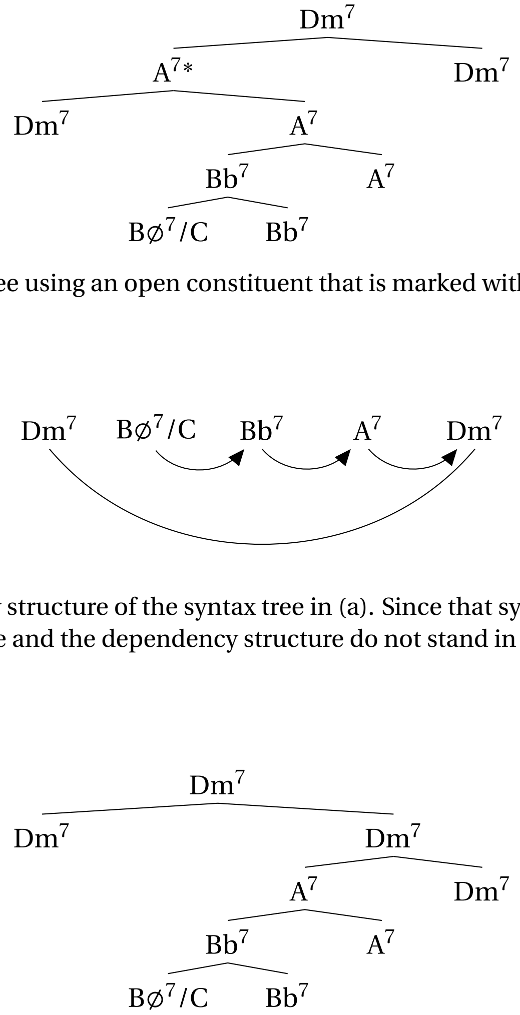
Constituents formalize the notion of a musical unit such as a chord or a phrase. In all derivation trees shown in the first chapter, the complete constituents are exactly the subsequences that are leafs of single subtrees. For instance, the A part of the tune *Take the A Train* — the chord sequence C $D^{7\sharp11}$ Dm^7 G^7 C — is a complete constituent. Its subsequence $D^{7\sharp11}$ Dm^7 G^7 is a complete constituent as well, but the subsequence C $D^{7\sharp11}$ Dm^7 is not. Formally, a subsequence is called a *complete constituent* if it contains a chord, called the *head*, that is transitively referred to by all other chords of the sequence. For instance, the chord $G^7$is the head of the phrase $D^{7\sharp11}$ Dm^7 G^7 and C is the head of the whole sequence C $D^{7\sharp11}$ Dm^7 G^7 C. In cases in which a constituent is embraced by a strong prolongation (e.g., for the whole sequence), the convention is used that the head is the right chord symbol. Since only the head of a complete constituent is allowed to refer to a chord outside the constituent, the concept of harmonic reference is generalizable to complete constituents: A complete constituent is defined to refer to a chord X if its head refers to X.

In addition to complete constituents, one other constituent type is used in the JHT analyses. Consider for example the first four measures of the Jazz standard *Why Don't You Do Right?*,

$$| \ Dm^7 \ \ B\varnothing^7/C \ | \ Bb^7 \ \ A^7 \ | \ Dm^7 \ \ B\varnothing^7/C \ | \ Bb^7 \ \ A^7 \ |,$$

where $B\varnothing^7/C$ denotes a half-diminished seventh chord with root B and a C in the bass. The first two measures constitute a phrase following the *Lamento* schema (a step-wise descending movement of the bass from scale degree I to scale degree V (Caplin, 2014)) that is repeated multiple times in the song. Since the transition from A^7 to Dm^7 does not sound like a resolution but more like a jump or an interruption (partly because of the repetition of the first two measures), A^7 is assumed to not resolve into the following tonic Dm^7, but into a tonic later in the song. Therefore, the phrase Dm^7 $B\varnothing^7/C$ Bb^7 A^7 does constitute some kind of unit as shown in Figure 4.1a.

Since Dm^7 and A^7 both refer to a chord outside the phrase (see Figure 4.1b), the phrase does not have a head. It is therefore not a complete constituent. Such constituents, in which multiple chords refer to a chord outside of the phrase, are called *open constituents*. The chords of an open constituent that refer to a chord outside of the constituent are called *chords with open references*. In the example of *Why Don't You Do Right?*, the chords Dm^7 and A^7 are the chords with open references of the open constituent Dm^7 $B\varnothing^7/C$ Bb^7 A^7. Both chords Dm^7 and A^7 refer to the same tonic chord Dm^7.



(a) Syntax tree using an open constituent that is marked with an asterisk.

(b) Harmonic dependency structure of the syntax tree in (a). Since that syntax tree contains an open constituent, the syntax tree and the dependency structure do not stand in 1-to-1 relation.

(c) Resolution of the open constituent in the syntax tree shown in (a). This tree stands in 1-to-1 relation to the dependency structure in (b).

Figure 4.1 – Hierarchical analysis of the initial chords of the Jazz standard *Why Don't You Do Right?* using open constituents (marked with asterisks). The last tonic chord Dm7 represents the end of a chorus. The conversion of the open constituents into a pure prolongation-preparation structure is shown in 4.1c.

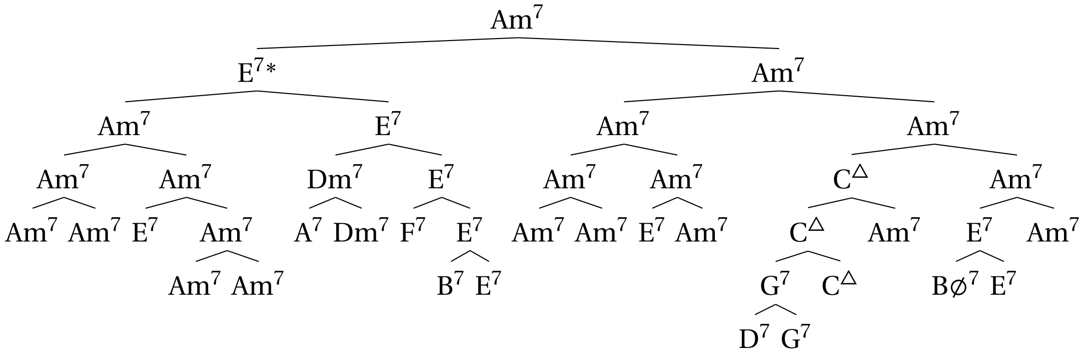

Am7
E7*
Am7
Am7
E7
Am7
Am7
Am7
Am7
Dm7
E7
Am7
Am7
C△
Am7
Am7 Am7 E7 Am7 A7 Dm7 F7 E7 Am7 Am7 E7 Am7 C△ Am7 E7 Am7
Am7 Am7 B7 E7 G7 C△ Bø7 E7
D7 G7

Figure 4.2 – Complete syntax tree of the Jazz standard *Summertime* (turnaround omitted). The top levels of the tree reflect the ABAC form the song using an open constituent governing the first two section AB.

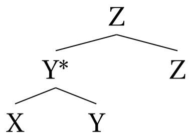
The JHT allows a single type of open constituent, called *restricted* open constituent, which consists of two adjacent constituents that refer to the same chord later in the sequence. Since all constituents considered in the JHT are restricted in that way, they are simply referred to as open constituents. The restriction enables a further generalization of harmonic reference to open constituents: an open constituent is defined to refer to the chord to which all of its chords with open references refer. As shown in Figure 4.1a, the topmost node of an open constituent is labeled by the chord symbol of the right child of the node and additionally marked with an asterisk.

Other examples of open constituents are (i) I-VI-II-V-like phrases in *I Got Rhythm* and *I Can't Give You Anything But Love* and, in particular, (ii) Jazz standards of form ABAC in which the B-part ends in a half cadence such as *All of Me*, *How High the Moon*, and *A Fine Romance*. The standard *Summertime*, shown in Figure 4.2, is a prototypical example of a song with a ABAC form and a half cadence at the end of the B section. The interruption after the half cadence is supported by the movement from scale degree 3 to scale degree 2 in the melody and denoted using an open constituent.

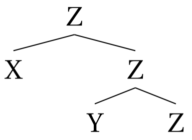
4.2.1 Interpretation of open constituents as prolongation-preparation structures

Syntax trees containing open constituents are interpretable as harmonic dependency structures as shown in Figure 4.1. The interpretation procedure transforms a syntax tree that contains open constituents (e.g., Figure 4.1a) in to a tree that only represents prolongation and preparation operations (e.g., Figure 4.1c). This transformed tree then characterizes the dependency structure (e.g., Figure 4.1b). Since open constituents are explicitly marked with asterisks, their interpretation is unambiguous.

To formalize the interpretation of open constituents, let Y^* be the chord symbol labeling an open constituent consisting of two constituents labeled with chord symbols X and Y. Let further be Z the chord symbol that is referenced by both X and Y. The reference is expressed by Z being the right sibling of the open constituent. The conversion then transforms



into



In the more general case of nested open constituents, the conversion is recursively applied from the root to the leaves of the tree (i.e., top-down).

Note that the definition of open constituents given above guarantees that both $Z \longrightarrow X \;\; Z$ and $Z \longrightarrow Y \;\; Z$ are rules of the harmony grammar. The transformation therefore always creates valid prolongation-preparation structures. The converse is not true in general. Since open constituents encode information about phrase structure, the transformation is lossy. A transformation of prolongation-preparation structures to constituent structures that describe the phrase structure of a sequence thus needs to take additional information into account such as the the tune's melody and harmonic rhythm.

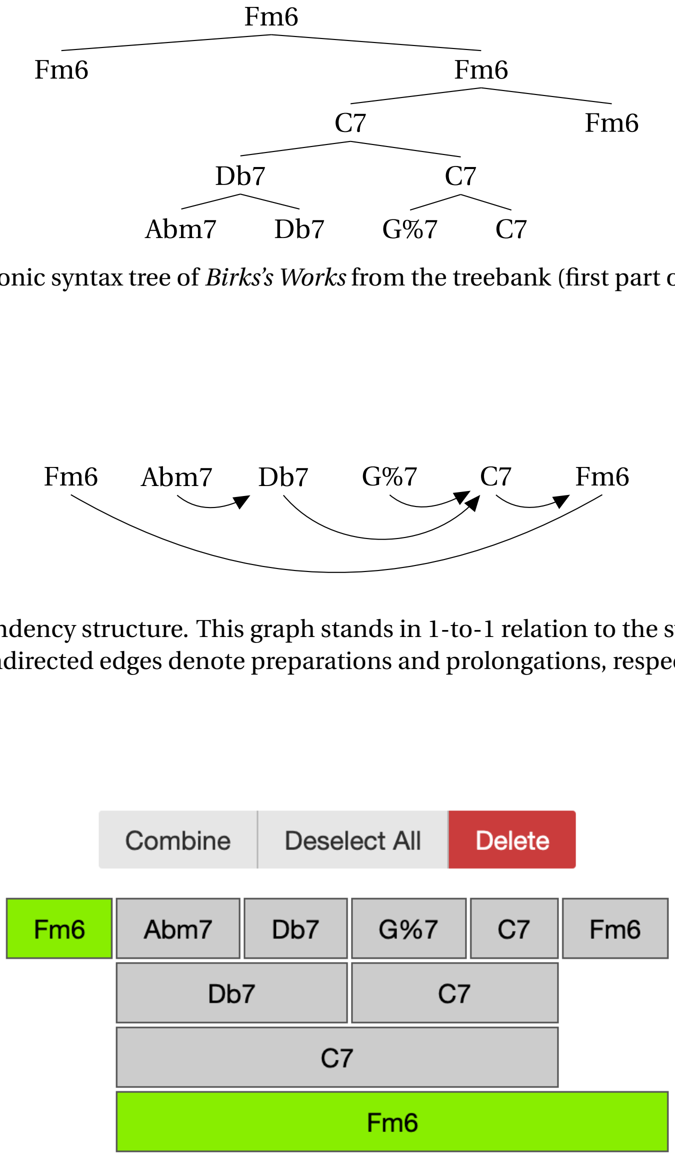
4.3 Tree Annotation Tool

The trees of the JHT are created using a graphical interface implemented as a simple web application, which was developed during the creation of the treebank. The source code of the application is written in ClojureScript (which compiles to JavaScript) and is publicly available on GitHub. The application itself is hosted on GitHub pages and can be used independently of this dataset.[5] A screenshot of the application is shown in Figure 4.3c. The main part of the user interface displays a syntax tree that is represented by a hierarchical button layout. The user interface also contains an input-output section, a preview of the annotated tree, and buttons for creating, deleting, and deselecting tree nodes.

To create a syntax tree, the user inputs a sequence of space-separated strings such as chord symbols. To create an inner node of the tree, the nodes that become the child nodes of the new inner node are selected and combined by pressing a button or a key shortcut. Since the trees are mostly right-headed, the label of the rightmost child is used for the new node by default, but the label of a node can be changed arbitrarily. The output of the application is given as a string representation of the tree in tikz-qtree format[6] as shown in Figure 4.3d. Existing trees can be edited by loading them in tikz-qtree or JSON format. Since the application is designed to be agnostic to annotation conventions, it allows arbitrary labels and rule arities.

[5] https://dcmlab.github.io/tree-annotation-code/

[6] https://www.ctan.org/pkg/tikz-qtree



(a) Part of the harmonic syntax tree of *Birks's Works* from the treebank (first part of a 5-part subfigure over 2 pages).

(b) Harmonic dependency structure. This graph stands in 1-to-1 relation to the syntax tree shown in (a). Directed and undirected edges denote preparations and prolongations, respectively.

(c) Screenshot of tree annotation app. Each button represents a tree node. The user is selecting the green buttons to combine them to the full tree.

```
[.Fm6
  Fm6
  [.Fm6
    [.C7
      [.Db7
        Abm7
        Db7 ]
      [.C7
        G\%7
        C7 ] ]
    Fm6 ] ]
```

(d) String representation of the syntax tree in tikz-qtree format. This string is created using the tree annotation app shown in (d). The tree plot is shown in (a).

```
{"label": Fm6, "children": [
  {"label": "Fm6", "children": []},
  {"label": "Fm6", "children": [
    {"label": "C7", "children": [
      {"label": "Db7", "children": [
        {"label": "Abm7", "children": []},
        {"label": "Db7", "children": []}]},
      {"label: "C7", "children": [
        {"label: "G%7", "children": []},
        {"label: "C7", "children": []}]}]},
    {"label": "Fm6", "children": []}]}]}
```

(e) Tree string in JSON format, automatically converted from tikz-qtree format shown in (c). The JHT stores trees in this format.

Figure 4.3 – Syntax tree of the final chords of the Jazz standard *Birk's works* in different representations.

4.4 Annotation Procedure

All analyses in the dataset begin from chord sequences drawn from the iRealPro collection of Jazz standards. This collection was created by the user community of the iRealPro app[7] and transferred into kern format by Shanahan and Broze (2012).[8] The data was transformed into a JSON-like format; individual chord symbols were occasionally corrected when significant differences between the iRealPro data and publicly available *Real Books* were noticed. Annotations of bass notes and optional chord tones such as ninths and elevenths were excluded from the chord symbols. 150 Jazz standards were selected for analysis (i) by filtering pieces that are within the scope of the treebank described at the beginning of this chapter and (ii) by preferring shorter pieces. If applicable, turnarounds at the end of lead sheets were deleted or a final tonic chord not contained in the lead sheet was added. All repetitions were unfolded and codas were appended at the positions indicated in the lead sheet. The selected Jazz standards were initially analyzed by Daniel Harasim and a student assistant. The analyses were then reviewed by Christoph Finkensiep and Petter Ericson, discussed in the group, and edited accordingly.

Every hierarchical analysis denotes at least one author's mental representation of the harmonic structure of a Jazz standard. Each analysis is therefore also influenced by other musical features such as harmonic rhythm, phrasing, musical form, and melody. In ambiguous cases, the analyst chose the option that he deemed most important. These choices were necessary, because a single syntax tree can only encode one harmonic function for each chord. For example, a C major triad can as a scale degree I act as a tonic in C major or as a scale degree V/IV act as the preparation of a following IV. The latter is more common in middle sections.

Since the iRealPro lead sheets were created and collected by the community of the application, the chord symbol usage is not fully consistent across the pieces. For instance, a Fm6 chord symbol can denote a tonic chord in F minor over a Dorian scale or a Bb9 chord with omitted root and fifth in the bass. Another example is that fourth-voicings are commonly denoted as suspension chords while actual suspensions of the scale degree V (e.g., G C E suspending G B D) are sometimes denoted as chords over the scale degree I (with or without explicitly mentioning the second inversion).

Furthermore, some chords do not have a proper harmonic function, but are better explained as voice-leading connections between two chords. The chords C $C\sharp^{\circ 7}$ G/D at the beginning of the final 8 measures of *Bill Bailey* are an example of such a voice-leading connection (see Figure 4.4). The final measures of the Jazz standard *Bill Bailey* are moreover an example of a common closing pattern. This pattern starts on the scale degree IV in its first measure, then transitions to a suspension of the scale degree V in measure 3, jumps away, and finally approaches the tonic through the cycle of fifths.

[7] https://irealpro.com/

[8] The iRealPro dataset is available in kern format at http://doi.org/10.5281/zenodo.3546040.

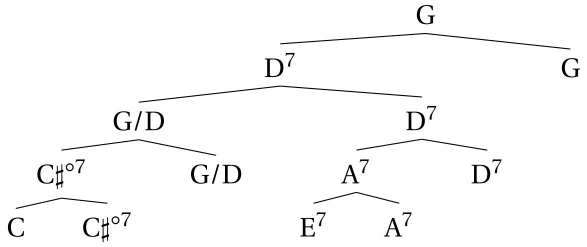


Figure 4.4 – Syntax tree of the final 8 measures of the Jazz standards *Bill Bailey* as analysed in the JHT (turnaround omitted).

4.5 Dataset Summary

The JHT is provided as a single file in JavaScript Object Notation (JSON) format. For each Jazz standard, this file contains the chord sequence with rhythmical information (measures and beats), metadata about title, composer(s), year of composition, time signature, and key[9] as well as the harmonic syntax trees as shown in Figure 4.2.

In addition to the hierarchical analyses, some pieces contain a turnaround annotation represented as an integer. A value of zero means that the Jazz standard ends with a tonic chord. A positive value n means that the lead sheet of the piece ends with a turnaround of length n. For example, the chord sequence of *I love Paris* is in C major and ends with the chords Dm^7 G^7 C^6 $\mathrm{D}\varnothing^7$ G^7. It therefore has a turnaround length of $n = 2$. A negative turnaround annotation means that the tonic of the piece is not at the end of the piece, but at the beginning. A value of -1 indicates, for example, that the first chord of the chord sequence is the tonic of the piece, like in *Solar*. In rare cases, the tonic is not the first chord but the n-th chord which is represented by a turnaround annotation of $-n$.

The 150 chord sequences analysed in the treebank have an average length of 27.75 and consist of 11697 chords in total with 92 unique chord symbols. The syntax trees consist in total of 3899 binary rule applications with 512 unique rules and 268 open constituents. The average tree height is 7.57.

Further descriptive statistics of the JHT are visualized in Figures 4.5–4.9. Figure 4.5 shows that the subset of the analyzed pieces is chosen relatively independently from the year of composition. Figure 4.6 shows the bias for short pieces in this subset. Figure 4.7 shows that the length of turnarounds, if present, usually ranges between 1 and 3.

Figures 4.8 and 4.9 show separately for major and minor keys how often a context-free grammar rule is used in the hierarchical analyses. For these plots, all chord sequences were transposed to C major or to C minor, respectively. Prolongations of the tonic, preparations of the tonic by the fifth scale degree, and preparations of the fifth scale degree by the second are by far the most common rules.

[9] This data and metadata was copied from the iRealPro dataset in kern format.

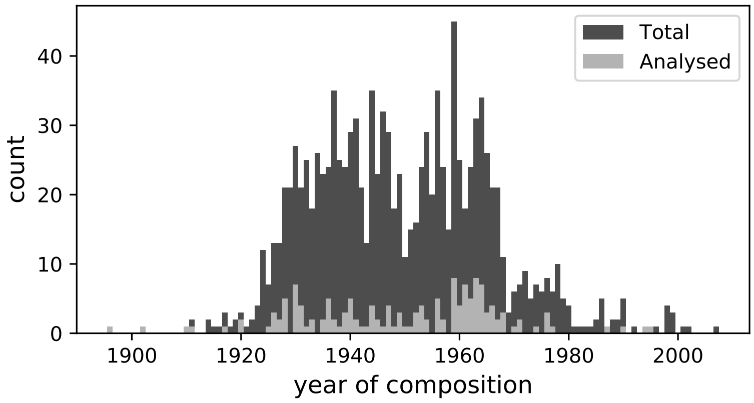


Figure 4.5 – Number of Jazz standards by year of composition as written in the iRealPro dataset.

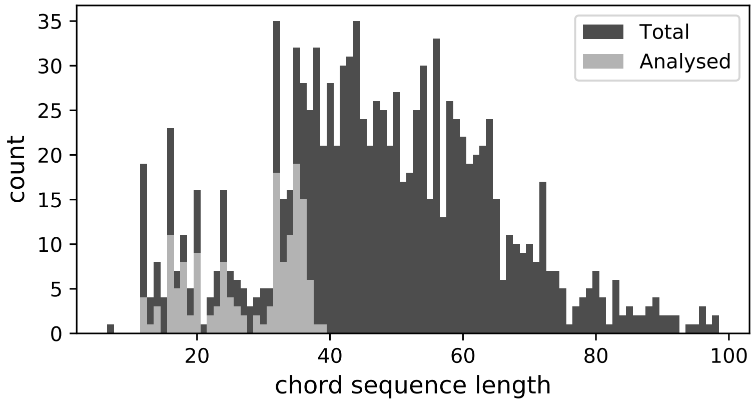


Figure 4.6 – Number of Jazz standards by chord sequence length. Sequences with more than 100 chords are omitted.

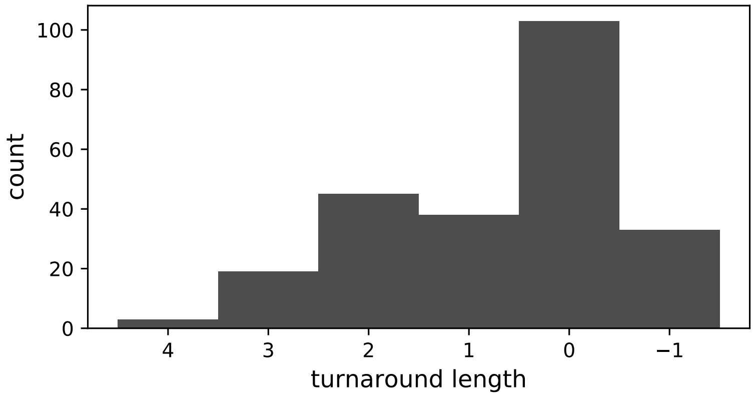


Figure 4.7 – Positive values indicate annotated turnaround lengths. A negative value of -1 indicates tunes with missing final tonic.

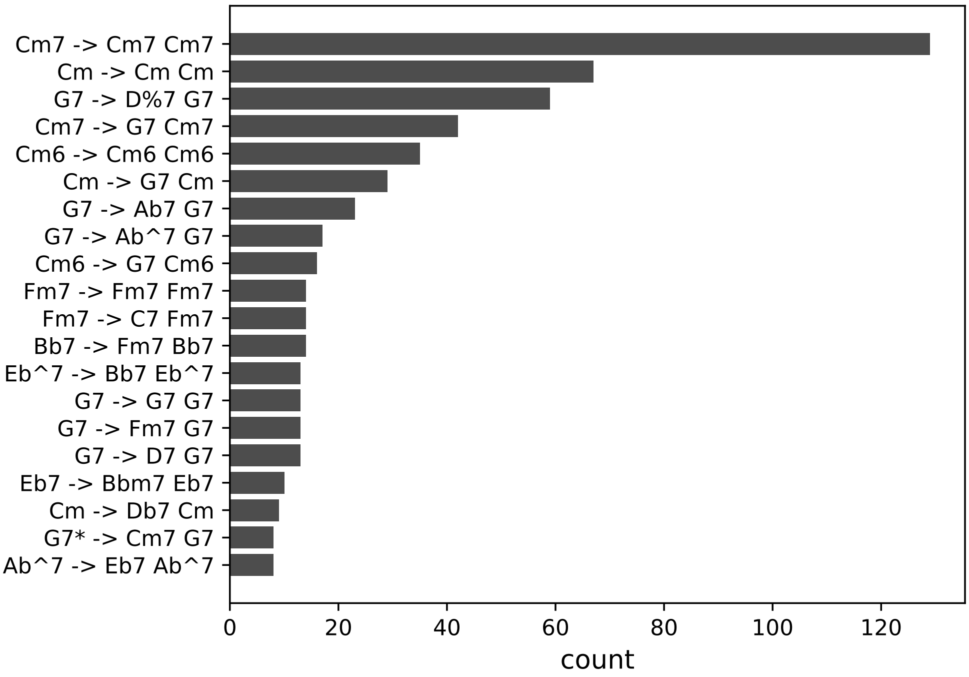


Figure 4.8 – 20 most frequent rules used to analyze tunes in minor keys. All minor tunes were transposed to C minor for this plot.

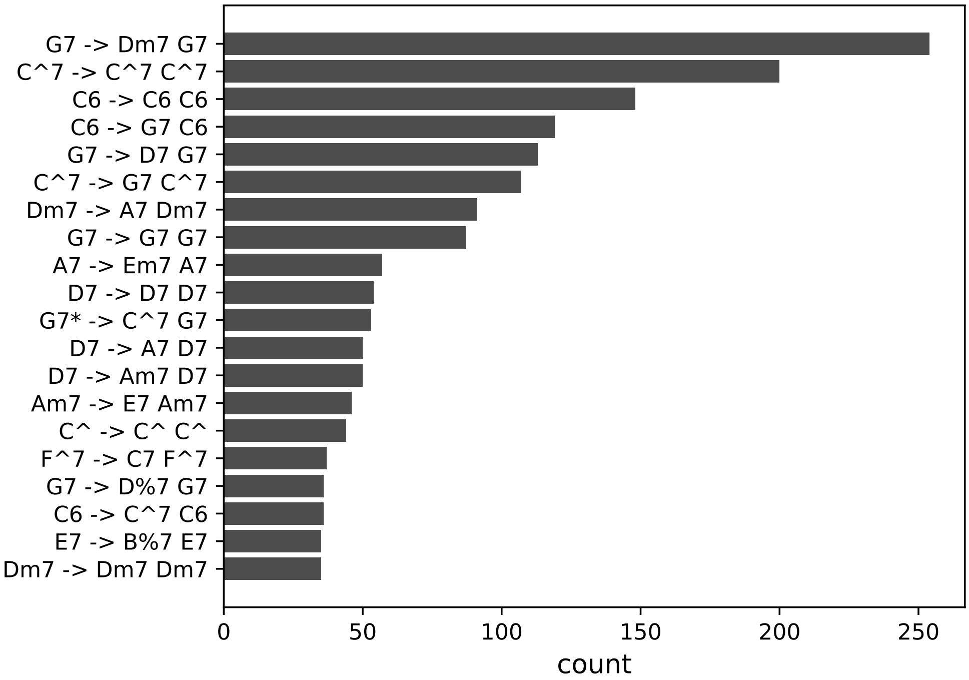


Figure 4.9 – 20 most frequent rules used to analyze tunes in major keys. All major tunes were transposed to C major for this plot.

5 Abstract context-free grammars[1]

This chapter presents Abstract Context-Free Grammars (ACFGs), a generalization of context-free grammars which allows for more flexible probabilistic modeling. The idea of ACFGs is crucial to this study since it enables the implementation of grammar models for harmony that go beyond the current state of the art and learn grammatical systems from the observation of chord sequences.

The chapter starts by giving a motivating example for what ACFGs will be used in the computational experiments. ACFGs and their derivation trees are then formally defined before the probabilistic version of an ACFGs, Probabilistic Abstract Context-Free Grammars (PACFGs) are presented. Afterwards, a construction of the product of two PACFGs is presented. Such product grammars are used in the computational experiments to implement joint grammar models for harmony and rhythm. The last section of this chapter finally gives an overview over related grammar formalisms and illustrates the formal expressiveness of ACFGs using an example.

5.1 Motivation: a grammar model for rhythm

Harmonic dependency structures are closely linked with *harmonic rhythm*, the rhythm in which the chords change in a sequence (Salley and Shanahan, 2016). Consider for example the harmonic derivation tree of the chord sequence of *Take the "A" train* shown in Figure 5.1. Additional to the chord symbols and scale degrees, this tree also shows the durations of its constituents (subtrees) as subscripts. The divisions of durations by binary rules correspond to simple split ratios. To concretize this fact, the *(left-)split ratio* of a binary rule application is defined as the proportion of the left child's duration relative to the parent's duration. For

[1] Abstract context-free grammars were originally proposed in a peer-reviewed article: Harasim, D., Rohrmeier, M., and O'Donnell, T. J. (2018). A Generalized Parsing Framework for Generative Models of Harmonic Syntax. *Proceedings of the 19th International Society for Music Information Retrieval Conference.* In a following article, the product grammar construction was introduced and two grammar models for rhythm were implemented: Harasim, D., O'Donnell, T. J., and Rohrmeier, M. (2019). Harmonic Syntax in Time: Rhythm Improves Grammatical Models of Harmony. *Proceedings of the 20th International Society for Music Information Retrieval Conference.*

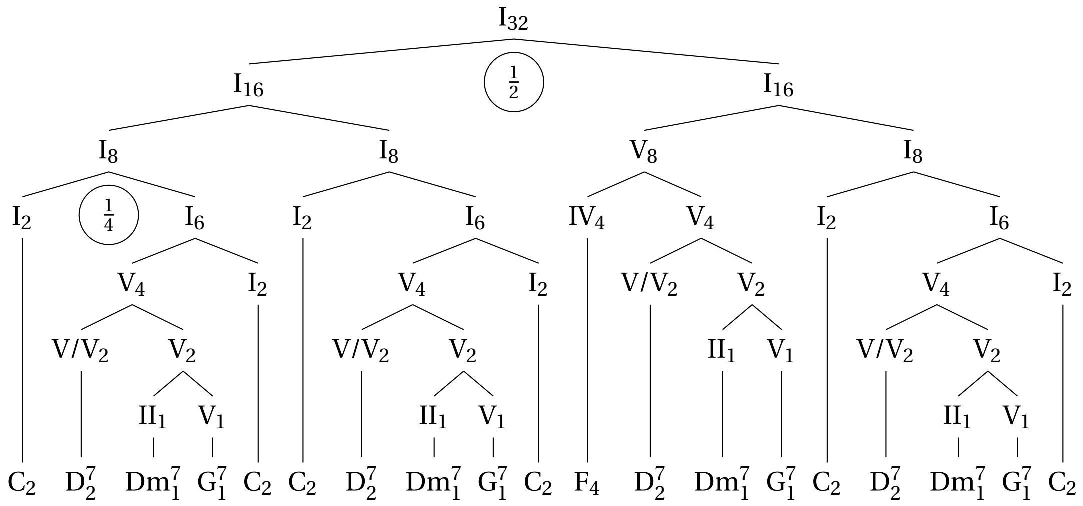

I_{32}
I_{16} $\frac{1}{2}$ I_{16}
I_8 I_8 V_8 I_8
I_2 $\frac{1}{4}$ I_6 I_2 I_6 IV_4 V_4 I_2 I_6
V_4 I_2 V_4 I_2 V/V_2 V_2 V_4 I_2
V/V_2 V_2 V/V_2 V_2 II_1 V_1 V/V_2 V_2
II_1 V_1 II_1 V_1 II_1 V_1
C_2 D^7_2 Dm^7_1 G^7_1 C_2 C_2 D^7_2 Dm^7_1 G^7_1 C_2 F_4 D^7_2 Dm^7_1 G^7_1 C_2 D^7_2 Dm^7_1 G^7_1 C_2

Figure 5.1 – Harmonic derivation tree of *Take the "A" train* including durations in measures as subscripts. The scale degrees are shown relative to C major. The circled fractions show two examples of left-split ratios.

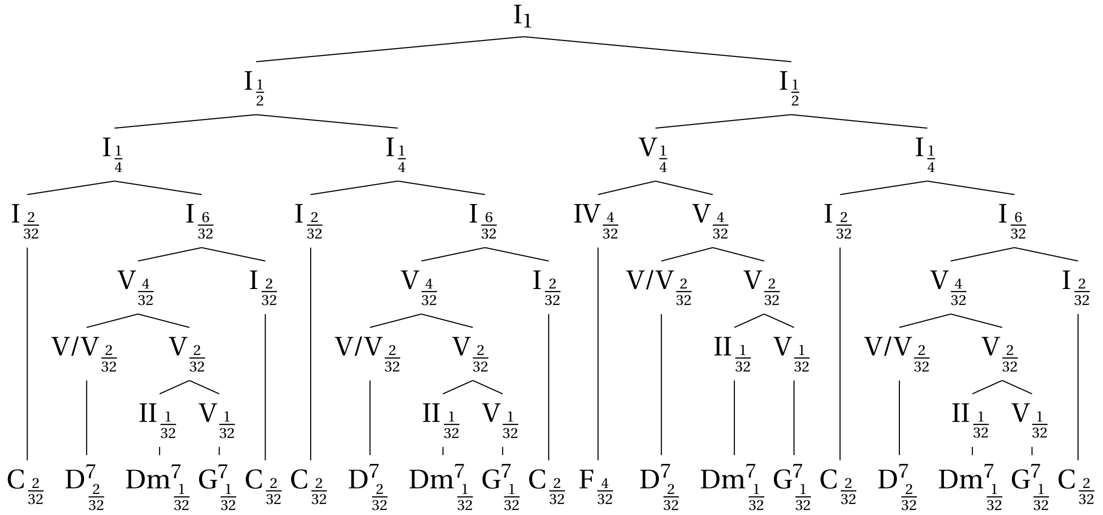

I_1
$I_{\frac{1}{2}}$ $I_{\frac{1}{2}}$
$I_{\frac{1}{4}}$ $I_{\frac{1}{4}}$ $V_{\frac{1}{4}}$ $I_{\frac{1}{4}}$
$I_{\frac{2}{32}}$ $I_{\frac{6}{32}}$ $I_{\frac{2}{32}}$ $I_{\frac{6}{32}}$ $IV_{\frac{4}{32}}$ $V_{\frac{4}{32}}$ $I_{\frac{2}{32}}$ $I_{\frac{6}{32}}$
$V_{\frac{4}{32}}$ $I_{\frac{2}{32}}$ $V_{\frac{4}{32}}$ $I_{\frac{2}{32}}$ $V/V_{\frac{2}{32}}$ $V_{\frac{2}{32}}$ $V_{\frac{4}{32}}$ $I_{\frac{2}{32}}$
$V/V_{\frac{2}{32}}$ $V_{\frac{2}{32}}$ $V/V_{\frac{2}{32}}$ $V_{\frac{2}{32}}$ $II_{\frac{1}{32}}$ $V_{\frac{1}{32}}$ $V/V_{\frac{2}{32}}$ $V_{\frac{2}{32}}$
$II_{\frac{1}{32}}$ $V_{\frac{1}{32}}$ $II_{\frac{1}{32}}$ $V_{\frac{1}{32}}$ $II_{\frac{1}{32}}$ $V_{\frac{1}{32}}$
$C_{\frac{2}{32}}$ $D^7_{\frac{2}{32}}$ $Dm^7_{\frac{1}{32}}$ $G^7_{\frac{1}{32}}$ $C_{\frac{2}{32}}$ $C_{\frac{2}{32}}$ $D^7_{\frac{2}{32}}$ $Dm^7_{\frac{1}{32}}$ $G^7_{\frac{1}{32}}$ $C_{\frac{2}{32}}$ $F_{\frac{4}{32}}$ $D^7_{\frac{2}{32}}$ $Dm^7_{\frac{1}{32}}$ $G^7_{\frac{1}{32}}$ $C_{\frac{2}{32}}$ $D^7_{\frac{2}{32}}$ $Dm^7_{\frac{1}{32}}$ $G^7_{\frac{1}{32}}$ $C_{\frac{2}{32}}$

Figure 5.2 – Harmonic derivation tree of *Take the "A" train* including durations as subscripts. The scale degrees are shown relative to C major and the durations are given relative to the duration of the whole chord sequence which is normalized to one.

instance, the split ratio of the root of the derivation tree is $\frac{16}{32} = \frac{1}{2}$ and the split ratio of the root of the A part is $\frac{2}{8} = \frac{1}{4}$. These split ratios are shown in the circles of Figure 5.1. The whole derivation tree uses 12 times the split ratio $\frac{1}{2}$, 3 times $\frac{1}{4}$, and 3 times $\frac{2}{3}$. Note that in particular the high levels of the tree, which describe the phrase structure of the sequence, use only the simplest split ratio $\frac{1}{2}$.

Since many harmonic phrases of Jazz standards are rhythmically regular, the modeling of rhythm improves the tree predictions of harmony grammars. It does so by resolving ambiguities that cannot be resolved from the chord symbols of a harmonic sequence alone. This is in particular shown in the computational experiments described in Chapters 9 and 9.

The C major chords in measures 7, 8, 9, and 10 of *Take the "A" train* (at the end of the first and the beginning of the second A part) represents a particularly ambiguous case. Without considering the first 8 measures a phrase of the tune, the C major triads in measures 7 to 10 could equally plausibly constitute a tonic prolongation. In the computational experiments of this study, joint grammar models for harmony and rhythm employed that are created using two separate *component grammars*, one grammar for harmony and one for rhythm. The following definition describes a naive formulation of the component grammar for rhythm as a probabilistic context-free grammar.

Naive probabilistic rhythm grammar The rhythm grammar uses positive rational numbers $0 < u \leq 1$ as terminals to represent chord durations relative to the entire chord sequence,

$$T = \{ u \in \mathbb{Q} \mid 0 < u \leq 1 \}. \tag{5.1}$$

The duration of the full sequence is normalized to one, which abstracts from the unit in which chord durations are measured. A derivation tree of *Take the "A" train* with normalized durations is shown in Figure 5.2. The set of nonterminals N essentially establishes a one-to-one correspondence between terminals and nonterminals. More precisely, N comprises one nonterminal for each terminal and an additional start symbol Start. Therefore, there is a bijection between the sets N and $T \uplus \{\text{Start}\}$, denoted by $N \cong T \uplus \{\text{Start}\}$. The sets T and N are kept disjoint to determine the termination of the grammar's generative process.

The set of rules R consists of one start rule Start $\longrightarrow$ 1, one terminal rule $u \longrightarrow \underline{u}$ for each $u \in N$ where the underline indicates a terminal, and one split rule $u \longrightarrow (su)\ (u - su)$ for for each $u \in N$ and finitely many split ratios $s \in S \subset \{ s \in \mathbb{Q} \mid 0 < s < 1 \}$. For each nonterminal $u \in N$, the set of applicable rules is thus:

$$R_u = \begin{cases} \{\text{Start} \longrightarrow 1\}, & \text{if } u = \text{Start} \\ \{u \longrightarrow \underline{u}\} \uplus \{u \longrightarrow (su)\ (u - su) \mid s \in S\}, & \text{otherwise} \end{cases} \tag{5.2}$$

The probability of a rule $r \in R_u$ being applied to a duration u is given by a categorical distribution over R_u whose parameters can be specified by hand or learned from data. Note that the usage of categorical distributions requires the finiteness of split ratios.

This naive formulation of the rhythm grammar has two problems. The first problem is less serious; It concerns the infinitely large sets of terminals and nonterminals. In practice, finite sets which contain all relevant — or if known all occurring — chord durations could be used instead. The second and more relevant problem is that the probability of a duration split depends on both the split ratio and the duration that is to be split. Music-theoretically, it makes, however, more sense to parameterize the probability independent from the actual duration so that it only depends on the split ratio, because the split ratio is independent of the lead-sheet notation of the tune. In the derivation of *Take the "A" train* shown in Figure 5.1 for example, the split ratio $\frac{1}{2}$ is commonly used both at the very top and the very bottom of the tree where it is applied to both the largest and the smallest constituent duration, respectively. A parameterization independent of the parent's duration increases moreover the robustness of the grammar model against uncommon durations.

To solve the described parameterization problem, this study proposes abstract context-free grammars which use partial functions as rules. A *partial function* $f\colon X \rightharpoonup Y$ is not required to be defined for all elements of X, but only for a subset $\mathrm{dom}(f) \subseteq X$ that is called the *domain* of f.[2] All binary rules of the rhythm grammar that have a common split ratio s can be "grouped" into a partial function

$$\textsc{split}_s\colon N \rightharpoonup N^2, \qquad \textsc{split}_s(u) = (su) \quad (u - su) \tag{5.3}$$

for which $\mathrm{dom}(\textsc{split}_s) = N \setminus \{\mathrm{Start}\}$. The terminal rules are expressed by a single partial function

$$\textsc{terminate}\colon N \rightharpoonup T, \qquad \textsc{terminate}(u) = \underline{u} \tag{5.4}$$

for which $\mathrm{dom}(\textsc{terminate}) = N \setminus \{\mathrm{Start}\}$. The start rule $\mathrm{Start} \longrightarrow 1$ can also be expressed as a partial function

$$\textsc{start}\colon N \rightharpoonup N, \qquad \textsc{start}(\mathrm{Start}) = 1 \tag{5.5}$$

with a *singleton* domain $\mathrm{dom}(\textsc{start}) = \{\mathrm{Start}\}$.

With these partial *rewrite functions*, the set R_u of rewrite functions that are applicable to a chord duration $u \in N$ does not depend on the value of u, but only on the fact whether u is the start symbol or not,

$$R_u = \begin{cases} \{\textsc{start}\}, & \text{if } u = \mathrm{Start} \\ \{\textsc{terminate}\} \uplus \{\textsc{split}_s \mid s \in S\}, & \text{otherwise} \end{cases} \tag{5.6}$$

[2] In a strictly typed programming language such as Haskell, partial functions $f: X \rightharpoonup Y$ are expressed by the typing `f :: X -> Maybe Y`.

To denote this independence, the function $\phi : N \to \{\text{Start}, \textsc{NotStart}\}$, called *nonterminal feature projection*, is used that checks whether a nonterminal is the start symbol or not:

$$\phi(u) = \begin{cases} \text{Start}, & \text{if } u = \text{Start} \\ \textsc{NotStart}, & \text{otherwise} \end{cases} \tag{5.7}$$

Since $\phi(u) = \phi(v)$ implies $R_u = R_v$ for all $u, v \in N$, all nonterminals u that are projected onto the same feature $\phi(u)$ can now share a categorical distribution over the rewrite functions R_u. In particular, this allows to parameterize the probabilities of split rules independent from the chord duration those rules are applied to. Moreover, the implication $\phi(u) = \phi(v) \implies R_u = R_v$ necessarily requires the set of nonterminals to be infinitely large.

5.2 Definition of abstract context-free grammars

Before the definition of abstract context-free grammars is stated, the following paragraphs introduce preliminary concepts and notation. The *image* of a partial function $f \colon X \nrightarrow Y$ is defined as the set of all values $y \in Y$ that are covered by f,

$$\text{image}(f) = \{y \in Y \mid \exists x \in \text{dom}(f) \colon f(x) = y\}. \tag{5.8}$$

The composition $g \circ f \colon X \nrightarrow Z$ of two partial functions $f \colon X \nrightarrow Y$ and $g \colon Y \nrightarrow Z$ is defined by

$$(g \circ f)(x) = \begin{cases} g(f(x)), & \text{if } x \in \text{dom}(f) \text{ and } f(x) \in \text{dom}(g) \\ \textsc{undefined}, & \text{otherwise.} \end{cases} \tag{5.9}$$

In particular, $\text{dom}(g \circ f) = \{x \in \text{dom}(f) \mid f(x) \in \text{dom}(g)\}$. A partial function $f \colon X \nrightarrow Y$ is a *partial bijection* if there exists a function $f^{-1} \colon \text{image}(f) \to \text{dom}(f)$ such that $f^{-1}(f(x)) = x$ for all $x \in \text{dom}(f)$ and $f(f^{-1}(y)) = y$ for all $y \in \text{image}(f)$.

The set of sequences consisting of elements from a set X is denoted by $X^* = \biguplus_{n \in \mathbb{N}} X^n$, where $X^0 = \{\varepsilon\}$ and ε denotes the empty sequence. The symbol $\uplus$ denotes the union of disjoint sets. The set of sequences excluding the empty list is denoted by $X^+ = X^* \setminus \{\varepsilon\}$. Sequences and vectors are usually written in bold font if they potentially contain more than one element. The length of a sequence $\boldsymbol{\alpha} \in X^*$ is denoted by $|\boldsymbol{\alpha}|$, the k-th element of $\boldsymbol{\alpha}$ is denoted by $\boldsymbol{\alpha}_k$ for $k \in \{1, \ldots, |\boldsymbol{\alpha}|\}$, and the subsequence from index k to index l is denoted by $\boldsymbol{\alpha}_{k:l}$. The set of sequences X^* is additionally equipped with the algebraic structure of a monoid.

Monoid Let X be a set. A function $\star : X \times X \to X$ is called an *associative binary operation* if for all $x, y, z \in X$, $x \star (y \star z) = (x \star y) \star z$. An element $x \in X$ is called the *identity element* of $\star$ if for all $y \in X$, $x \star y = y = y \star x$. A *monoid* $(X, \star, 1)$ consists of a set X, an associative binary operation $\star : X \times X \to X$, and an identity element $1 \in X$.

The set of sequences X^* forms a monoid $(X^*, \star, \varepsilon)$ with the empty sequence as the identity element and concatenation as the binary operation,

$$(\boldsymbol{\alpha} \star \boldsymbol{\beta})_k = \begin{cases} \boldsymbol{\alpha}_k, & \text{if } k \leq |\boldsymbol{\alpha}| \\ \boldsymbol{\beta}_{k-|\boldsymbol{\alpha}|}, & \text{if } k > |\boldsymbol{\alpha}| \end{cases} \tag{5.10}$$

for sequences $\boldsymbol{\alpha}, \boldsymbol{\beta} \in X^*$ and indices $k \in \{1, \dots, |\boldsymbol{\alpha}| + |\boldsymbol{\beta}|\}$. In the following, sequence concatenation is also denoted by $\boldsymbol{\alpha\beta} = \boldsymbol{\alpha} \star \boldsymbol{\beta}$.[3]

Abstract context-free grammar An *abstract context-free grammar* $G = (T, N, \text{Start}, R)$ consists of a set T of *terminals*, a set N of *nonterminals* disjoint to T, an initial nonterminal $\text{Start} \in N$, and a finite set R of partial bijections that map nonterminals to lists of terminals and nonterminals.[4] For each individual rule, those lists are additionally required to have equal length,

$$R \subset \{r \colon N \rightharpoonup (T \uplus N)^* \mid r \text{ bijective and } \exists n \in \mathbb{N} \colon r \colon N \rightharpoonup (T \uplus N)^n\}. \tag{5.11}$$

The elements of R are called the *rewrite rules* or *rewrite functions* of the grammar G. Let $A \in N$ denote an arbitrary nonterminal. The set of rules that are applicable to A is denoted by R_A,

$$r \in R_A \iff A \in \text{dom}(r). \tag{5.12}$$

For each rewrite rule $r \in R_A$, the constant length of $r(A)$ is called the *arity* of r and denoted by $\text{ar}(f)$. In the following, *Abstract Context-Free Grammar* is abbreviated by ACFG.

This definition is a generalization of the standard formulation of context-free grammars which can be recovered as a special case by requiring T and N to be finite sets and all domains of the rewrite functions to be sets containing exactly one element, $|\text{dom}(r)| = 1$ for all $r \in R$. Every statement presented for ACFGs therefore also applies to context-free grammars which are also referred to as *standard context-free grammars* in the following. While ACFGs do not require the set of terminals and nonterminal to be finite, the set of rules is, however, still required to be finite.

Abstract context-free grammars are called *abstract*, because their rewrite functions can explicitly represent abstract concepts such as prolongation or different kinds of preparation. To illustrate the definition and the notation of abstract context-free grammars, consider the following simplified harmony grammar with the seventh-chords of C major as terminals,

$$T = \{\text{C}^{\triangle}, \text{Dm}^7, \text{Em}^7, \text{F}^{\triangle}, \text{G}^7, \text{Am}^7, \text{B}\varnothing^7\}, \tag{5.13}$$

[3] The sequence monoid $(X^*, \star, \varepsilon)$ is also called the list monoid or the free monoid.

[4] The bijection requirement might be too strong in general. It is used in this thesis because it simplifies the definition of parsing algorithms by making use of the reversed rewrite functions.

scale degrees as nonterminals (represented as integers modulo 7),

$$N = \{\text{Start}, \text{I}, \text{II}, \text{III}, \text{IV}, \text{V}, \text{VI}, \text{VII}\}, \tag{5.14}$$

and rules $R = \{\textsc{start}, \textsc{prepare}, \textsc{prolong}, \textsc{terminate}\}$ for

$$\textsc{start}(X) = \begin{cases} \text{I}, & \text{if } X = \text{Start} \\ \text{undefined}, & \text{otherwise} \end{cases} \tag{5.15}$$

$$\textsc{prepare}(X) = X +_7 4 \qquad X \tag{5.16}$$

$$\textsc{prolong}(X) = X \qquad\qquad X \tag{5.17}$$

$$\textsc{terminate}(X) = \begin{cases} \text{C}^{\triangle}, & \text{if } X = \text{I} \\ \text{Dm}^7, & \text{if } X = \text{II} \\ \text{Em}^7, & \text{if } X = \text{III} \\ \text{F}^{\triangle}, & \text{if } X = \text{IV} \\ \text{G}^7, & \text{if } X = \text{V} \\ \text{Am}^7, & \text{if } X = \text{VI} \\ \text{B}\varnothing^7, & \text{if } X = \text{VII} \end{cases} \tag{5.18}$$

where the symbol $+_7$ denotes the addition modulo 7.

Note that this abstract context-free grammar has four rewrite functions whereas the corresponding standard context-free grammar has 22 rewrite rules. The derivation tree of the chord sequence $\text{C}^{\triangle}$ Am^7 Dm^7 G^7 $\text{C}^{\triangle}$ is shown in Figure 5.3.

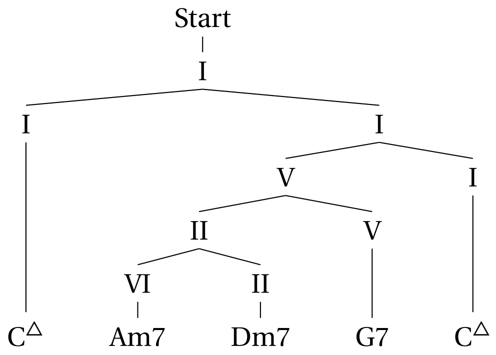


Figure 5.3 – Harmonic syntax tree using a simplified grammar.

$$
\begin{aligned}
\text{Start} &\longrightarrow_{\textsc{start}} && \text{I} \\
&\longrightarrow_{\textsc{prolong}} && \text{I} \quad \text{I} \\
&\longrightarrow_{\textsc{terminate}} && \text{C}^{\triangle} \quad \text{I} \\
&\longrightarrow_{\textsc{prepare}} && \text{C}^{\triangle} \quad \text{V} \quad \text{I} \\
&\longrightarrow_{\textsc{prepare}} && \text{C}^{\triangle} \quad \text{II} \quad \text{V} \quad \text{I} \\
&\longrightarrow_{\textsc{prepare}} && \text{C}^{\triangle} \quad \text{VI} \quad \text{II} \quad \text{V} \quad \text{I} \\
&\longrightarrow_{\textsc{terminate}} && \text{C}^{\triangle} \quad \text{Am}^{7} \quad \text{II} \quad \text{V} \quad \text{I} \\
&\longrightarrow_{\textsc{terminate}} && \text{C}^{\triangle} \quad \text{Am}^{7} \quad \text{Dm}^{7} \quad \text{V} \quad \text{I} \\
&\longrightarrow_{\textsc{terminate}} && \text{C}^{\triangle} \quad \text{Am}^{7} \quad \text{Dm}^{7} \quad \text{G}^{7} \quad \text{I} \\
&\longrightarrow_{\textsc{terminate}} && \text{C}^{\triangle} \quad \text{Am}^{7} \quad \text{Dm}^{7} \quad \text{G}^{7} \quad \text{C}^{\triangle}
\end{aligned}
$$

Figure 5.4 – Sequence of rule applications that generates the tree shown in Figure 5.3.

5.3 Trees as leftmost derivations and partial functions

This section describes a mathematical representation of derivation trees as rule sequences. This representation is then in particular used to characterize the *language* of an abstract context-free grammar G, that is the set of terminal sequences that G generates.

A rewrite rule is defined as a partial function $r\colon N \rightharpoonup (T \uplus N)^*$. Each rewrite function can thus be extended to a partial function $r\colon (T \uplus N)^* \rightharpoonup (T \uplus N)^*$ by applying r to the leftmost nonterminal of any input sequence $\boldsymbol{\alpha} \in (T \uplus N)^*$,

$$r(\boldsymbol{\alpha}) = \begin{cases} \boldsymbol{\alpha}' r(A)\boldsymbol{\alpha}'', & \text{if } \exists \boldsymbol{\alpha}' \in T^*, A \in \operatorname{dom}(r), \boldsymbol{\alpha}'' \in (T \uplus N)^* \colon \boldsymbol{\alpha} = \boldsymbol{\alpha}' A \boldsymbol{\alpha}'' \\ \text{UNDEFINED}, & \text{otherwise} \end{cases} \tag{5.19}$$

Note that this extension uses a slight overloading of notation for partial function application. This should not be confusing, because nonterminals can always be interpreted as singleton lists of nonterminals. With the proposed extension, sequences of rules $\boldsymbol{r} = r_1 \dots r_n \in R^*$ $(n \in \mathbb{N})$ are considered partial functions from $(T \uplus N)^*$ to $(T \uplus N)^*$ by partial function composition,

$$\boldsymbol{r}(\boldsymbol{\alpha}) = (\boldsymbol{r}_n \circ \dots \circ \boldsymbol{r}_1)(\boldsymbol{\alpha}). \tag{5.20}$$

As a special case, the empty rule sequence ε is applicable to all sequences $\boldsymbol{\alpha} \in (T \uplus N)^*$ by $\varepsilon(\boldsymbol{\alpha}) = \boldsymbol{\alpha}$.

Leftmost derivation A sequence of rules $\boldsymbol{r} \in R^*$ is called a *(leftmost) derivation* of $\boldsymbol{\beta} \in (T \uplus N)^*$ from $\boldsymbol{\alpha} \in (T \uplus N)^*$ if $\boldsymbol{r}(\boldsymbol{\alpha}) = \boldsymbol{\beta}$. Since only leftmost derivations are considered in this study, they are simply referred to as *derivations*. The set of all derivations of $\boldsymbol{\beta}$ from $\boldsymbol{\alpha}$ is denoted by $\text{DER}(\boldsymbol{\alpha}, \boldsymbol{\beta})$. The shorthand notation $\text{DER}(\boldsymbol{\beta}) := \text{DER}(\text{Start}, \boldsymbol{\beta})$ is used to refer to derivations from the start symbol.

For example, a derivation of the chord sequence $\text{C}^{\triangle}$ Am^7 Dm^7 G^7 $\text{C}^{\triangle}$ is shown in Figure 5.4. If the the rewrite rules do not overlap, each derivation tree of a terminal sequences has exactly one corresponding leftmost derivation. In contrast, a derivation tree with one or more nonterminal symbols at the leafs might not have a corresponding leftmost derivation. This is for example the case for sequences that start with a nonterminal and end with a terminal.

The language of the grammar G is the set of terminal sequences that have a derivation from its start symbol,

$$\text{LANGUAGE}(G) = \{\boldsymbol{\alpha} \in T^* \mid \text{DER}(\boldsymbol{\alpha}) \neq \emptyset\}. \tag{5.21}$$

5.4 Probabilistic abstract context-free grammars

Given a sequence of chords and a grammar that is able to generate this sequence, the task of inferring a derivation tree of the sequence is highly ambiguous in practice. Probabilistic grammars address this problem by assigning a probability to each rule. The product of the rule probabilities of a derivation is then interpreted as the derivation's plausibility.

Probabilistic abstract context-free grammar A *Probabilistic Abstract Context-Free Grammar* (PACFG), or short *probabilistic grammar* is an abstract context-free grammar where each nonterminal $A \in N$ is associated with a distribution over rewrite rules such that the probability of a rule $r \in R$ is positive if and only if r can be applied to A,

$$p_A(r) > 0 \iff A \in \operatorname{dom}(r). \tag{5.22}$$

The probability $p_{\boldsymbol{\alpha}}(r)$ of applying a rule $r \in R$ to a sequence $\boldsymbol{\alpha} \in (T \uplus N)^*$ is defined as the probability of r being applied to the leftmost nonterminal of α,

$$p_{\boldsymbol{\alpha}}(r) = p_{\text{LEFTMOST}(\boldsymbol{\alpha})}(r), \tag{5.23}$$

where

$$\text{LEFTMOST}(\boldsymbol{\alpha}) = \begin{cases} A, & \text{if } \exists \boldsymbol{\alpha}' \in T^*, A \in \operatorname{dom}(r), \boldsymbol{\alpha}'' \in (T \uplus N)^* : \boldsymbol{\alpha} = \boldsymbol{\alpha}' A \boldsymbol{\alpha}'' \\ \text{UNDEFINED}, & \text{otherwise} \end{cases} \tag{5.24}$$

and $p_{\text{UNDEFINED}}(r) = 0$. A probabilistic grammar is called *consistent* if the probability of its language is one,

$$\sum_{\boldsymbol{\alpha} \in T^*} \sum_{\boldsymbol{r} \in \text{DER}(\boldsymbol{\alpha})} \prod_{k=1}^{|\boldsymbol{r}|} p_{\boldsymbol{r}_{1:k-1}(\text{Start})}(\boldsymbol{r}_k) = 1. \tag{5.25}$$

For clarification, the product over rewrite probabilities can also be written as

$$\prod_{k=1}^{|\boldsymbol{r}|} p_{\boldsymbol{r}_{1:k-1}(\text{Start})}(\boldsymbol{r}_k) = p_{\text{Start}}(\boldsymbol{r}_1) \cdot p_{\boldsymbol{r}_1(\text{Start})}(\boldsymbol{r}_2) \cdot p_{\boldsymbol{r}_2(\boldsymbol{r}_1(\text{Start}))}(\boldsymbol{r}_3) \cdot \ldots \cdot p_{\boldsymbol{r}_{1:|\boldsymbol{r}|-1}(\text{Start})}(\boldsymbol{r}_{|\boldsymbol{r}|}). \tag{5.26}$$

Probabilistic grammars can be used to randomly generate derivations and sequences of terminals as described in Algorithm 1. Naively, one might think that all probabilistic grammars are consistent. Why should they not be? The reason is that it is in general not guarantied that the while loop in Algorithm 1 terminates. If this is the case, then the output of the algorithm can intuitively be understood as an infinitely long derivation. Then the grammar puts probability mass on infinite sequences of terminals and the probability of the grammar's language is strictly smaller than one.

Algorithm 1 Sample sequence from probabilistic grammar

```
    Input: probabilistic abstract context-free grammar $G = (T, N, \text{Start}, R)$
    Output: sequence $\boldsymbol{\alpha} \in T^*$, derivation $\boldsymbol{r} \in \textsc{Der}(\alpha) \subset R^*$

$\boldsymbol{\alpha} \leftarrow \text{Start} \in (T \uplus N)^*$
$\boldsymbol{r} \leftarrow \varepsilon \in R^*$
while $\boldsymbol{\alpha} \notin T^*$ do                         ▷ while $\boldsymbol{\alpha}$ contains nonterminals
    $A \leftarrow$ leftmost nonterminal of $\boldsymbol{\alpha}$
    sample rule $r$ according to $p_A(r)$
    $\boldsymbol{\alpha} \leftarrow r(\boldsymbol{\alpha})$
    $\boldsymbol{r} \leftarrow \boldsymbol{r}\, r$                                ▷ append rule $r$ to derivation $\boldsymbol{r}$
end while
```

A simple example of a grammar for which Algorithm 1 does not always terminate is the standard context-free grammar with only one terminal, $T = \{a\}$, two nonterminals $N = \{\text{Start}, A\}$, and three rules $\text{Start} \longrightarrow A$, $A \longrightarrow A\ A$, and $A \longrightarrow a$ such that $p(\text{Start} \longrightarrow A) = 1$, $p(A \longrightarrow A\ A) = q$, and $p(A \longrightarrow a) = 1 - q$ for some probability $q > 0.5$.[5] In fact, the probability that Algorithm 1 terminates for this particular grammar is given by the probability of the grammar's language,

$$\begin{aligned}
\sum_{\boldsymbol{\alpha}\in T^+}\sum_{\boldsymbol{r}\in\textsc{Der}(\boldsymbol{\alpha})}\prod_{k=1}^{|\boldsymbol{r}|} p_{\boldsymbol{r}_{1:k-1}(\text{Start})}(\boldsymbol{r}_k) &= \sum_{\boldsymbol{\alpha}\in T^+} C_{|\boldsymbol{\alpha}|-1}\, q^{|\boldsymbol{\alpha}|-1}(1-q)^{|\boldsymbol{\alpha}|} && (5.27)\\
&= \sum_{n=1}^{\infty} C_{n-1}\, q^{n-1}(1-q)^{n} && (5.28)\\
&= \begin{cases} 1, & \text{if } 0 < q \le 0.5\\ \frac{1-q}{q}, & \text{if } 0.5 < q \le 1 \end{cases} && (5.29)
\end{aligned}$$

where $C_{n-1} = \frac{1}{n}\binom{2n-2}{n-1}$ is the $n-1$-th *Catalan number*, i.e. the number of binary trees with n leafs. Equation 5.27 uses the fact that a sequence of length n must be generated by $n-1$ applications of the rule $A \longrightarrow A\ A$ and n applications of the rule $A \longrightarrow a$. Equation 5.28 uses the characterization of sequences over one symbol by their lengths — the unary encoding of natural numbers. The calculation of the infinite series in Equation 5.29 was verified using WolframAlpha.[6] For $q = \frac{2}{3}$, the probability of termination is 0.5. Therefore, one half of the probability mass is assigned to finite sequences and the other half is assigned to the infinite sequence of terminals a.

The presented example is particularly relevant for musical grammars because of the general principle of prolongation. If the rule probabilities of a grammar are assigned by hand, the issue of potential inconsistency should be considered; All prolongation rules should have a probability of less than 0.5.

[5] For probabilities of standard context-free rules, the notation $p(A \longrightarrow \boldsymbol{\alpha}) := p_A(A \longrightarrow \boldsymbol{\alpha})$ is used.

[6] The series $\sum_{n=0}^{\infty} C_n(q-q^2)^n = q^{-1}$ for $0.5 < q < 1$ was calculated on https://www.wolframalpha.com.

Booth and Thompson (1973) presented a general criterion to test a probabilistic standard context-free grammar for consistency (see Wetherell (1980) for a review of stochastic properties of probabilistic standard context-free grammars). One considers the square matrix $E \in \mathbb{R}^{n\times n}$ of expected nonterminal counts of right-hand sides of rules,

$$E_{B,A} = \mathbb{E}\left[\#(B \in r(A))\right] = \sum_{r\in R_A} p_A(r)\,\#(B \in r(A)) = \sum_{r\in R_A} p_A(r) \sum_{j=1}^{|r(A)|} \mathbb{1}\big(B = r(A)_j\big), \tag{5.30}$$

and calculates the *spectral radius* of E, which is in the case of a finite number of nonterminals equal to the maximal absolute value of the Eigenvalues of E. If the spectral radius is less than one, then the grammar is consistent. If it is greater than or equal to one, it could be either consistent or not. In practice however, probabilistic grammars induced from data are found to be consistent. Chi and Geman (1998) proved moreover that the special case of the maximum-likelihood estimate of a standard context-free grammar always yields a consistent grammar.

In the case of an infinite number of nonterminals, the matrix of expected values can be substituted by a continuous linear mapping

$$E\colon \ell^1 \to \ell^1, \quad E(x)_B = \sum_{A\in N} \mathbb{E}\left[\#(B \in r(A))\right] x(A), \tag{5.31}$$

where $\ell^1 = \{x\colon \mathbb{N} \to \mathbb{R} \mid \sum_{A\in N} |x(A)| < \infty\}$. We conjecture that the grammar is then consistent if the spectral radius of the mapping E is less than one. The spectral radius might, however, be more difficult to compute than in the finite case, because it is not characterized by Eigenvalues. The proof of this sufficient condition of consistency is left for future research.

In the rest of this study, all probabilistic grammars are assumed to be consistent. Then, the probability of the application of a derivation $\boldsymbol{r} \in R^*$ to a sequence $\boldsymbol{\alpha} \in (T \uplus N)^*$ is the product of the derivations rule applications,

$$p_{\boldsymbol{\alpha}}(\boldsymbol{r}) = \prod_{k=1}^{|r|} p_{\boldsymbol{r}_{1:k-1}(\boldsymbol{\alpha})}(\boldsymbol{r}_k), \tag{5.32}$$

and $p(\boldsymbol{r}) = p_{\text{Start}}(\boldsymbol{r})$. The probability that a sequence of terminals $\boldsymbol{\beta} \in T^*$ is derived from a sequence $\boldsymbol{\alpha} \in (T \uplus N)^*$ is the sum of the derivations' probabilities,

$$p_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) = \sum_{\boldsymbol{r}\in \textsc{Der}(\boldsymbol{\beta})} p_{\boldsymbol{\alpha}}(\boldsymbol{r}), \tag{5.33}$$

and $p(\boldsymbol{\beta}) = p_{\text{Start}}(\boldsymbol{\beta})$.

5.5 Product grammars

This study uses joint grammar models of harmony and rhythm called *product grammars*. These product grammars are constructed from two *component grammars*: a grammar for harmony and a grammar for rhythm. Intuitively speaking, the derivation trees of the product grammar are the tree structures that have derivations in both component grammars. The product of the probabilities of these component-grammar derivations then defines the probability of the corresponding product-grammar derivation. The mathematical idea of product grammars is similar to the idea of coupled-context-free grammars (Pitsch, 1994; Hotz and Pitsch, 1996).

Product grammar Let $G = (T, N, \text{Start}, R)$ and $G' = (T', N', \text{Start}', R')$ be two PACFGs, called *component grammars*. The product grammar

$$G \bowtie G' = (T \times T', N \times N', (\text{Start}, \text{Start}'), R \bowtie R') \tag{5.34}$$

is constructed from the Cartesian products of the sets of terminals and nonterminals, and the pair of the start symbols. The rewrite functions of $G \bowtie G'$ are the pairs of rewrite functions of equal arity,

$$R \bowtie R' = \{(r, r') \in R \times R' \mid \text{ar}(r) = \text{ar}(r')\}. \tag{5.35}$$

The application of a product rule $(r, r') \in R \times R'$ to a product nonterminal $(A, A') \in N \times N'$ is defined component-wise,

$$(r, r')(A, A') = \textsc{zip}(r(A), r'(A')), \tag{5.36}$$

where ZIP is the canonical transformation of a pair of sequences into a sequence of pairs,

$$\textsc{zip}\colon \{(\boldsymbol{x}, \boldsymbol{y}) \in X^* \times Y^* \mid |\boldsymbol{x}| = |\boldsymbol{y}|\} \to (X \times Y)^*, \qquad (\textsc{zip}(\boldsymbol{x}, \boldsymbol{y}))_j = (\boldsymbol{x}_j, \boldsymbol{y}_j). \tag{5.37}$$

The probability of a product rule application is proportional to the product of the probabilities of the rule application components,

$$p_{(A,A')}((r, r')) \propto p_A(r)\ p_{A'}(r'). \tag{5.38}$$

The reason why the product of the component probabilities does not yield a normalized probability distribution is that only rules of equal arity form valid product rules. Because of this requirement, the component rules r and r' cannot be sampled independently. For example, if G and G' are probabilistic standard context-free grammars with $R_A = \{A \longrightarrow A\ A\}$ and $R'_{A'} = \{A' \longrightarrow A'\ A',\ A' \longrightarrow a'\}$ as the sets of rules applicable to A and A', respectively, then $(A \longrightarrow A\ A, A' \longrightarrow A'\ A')$ is the only product rule applicable to (A, A'). However, the product of the probabilities of applying $A \longrightarrow A\ A$ to A and $A' \longrightarrow A'\ A'$ to A' does not equal 1:

$$\begin{aligned}
& p_A(A \longrightarrow A\ A)\ p_{A'}(A' \longrightarrow A'\ A') \\
& < p_A(A \longrightarrow A\ A)\ p_{A'}(A' \longrightarrow A'\ A') + p_A(A \longrightarrow A\ A)\ p_{A'}(A' \longrightarrow a') && (5.39) \\
& = p_A(A \longrightarrow A\ A)\big(p_{A'}(A' \longrightarrow A'\ A') + p_{A'}(A' \longrightarrow a')\big) && (5.40) \\
& = 1 && (5.41)
\end{aligned}$$

The grammar-learning methods used in the computational experiments presented in the third part of this thesis do not require normalizing constants of rule distributions to be calculated explicitly. Instead, only computations of expected values are needed which are approximated via Monte-Carlo methods.

5.6 Expressive power of ACFGs and grammatical formalisms

This section gives an overview over related grammar formalisms and illustrates the expressive power of ACFGs using the example of the copy language. It is, however, not essential for this study's argumentation.

Since the syntax of natural language exhibits richer structure than standard context-free grammars can express, much research on grammatical formalisms focused on extending the expressiveness of standard context-free grammars. Such formalisms are called *mildly context-sensitive*, because they require some but not all of the power of the much more expressive class of context-sensitive grammars. This restriction is desireable, because less expressive formalisms generally enable asymptotically faster computation algorithms. Examples of mildly context-sensitive formalisms are tree-adjoining grammars (Joshi et al., 1975; Joshi, 1985; Joshi and Schabes, 1997), linear context-free rewriting systems (Vijay-Shanker et al., 1987), multiple context-free grammars (Seki et al., 1991), and derivational minimalism (Stabler, 1996, 2011). Some of these grammatical frameworks are *weakly equivalent*; they are capable of expressing the same class of languages, but possibly with differently structured derivations (Weir, 1988; Joshi et al., 1991; Vijay-Shanker and Weir, 1994; Michaelis, 1998).

The structure of ACFGs is similar to the structure of definite clause grammars, another related formalism which allows structured and infinitely many nonterminals, but does not use bijective rewrite functions (Colmerauer, 1978; Pereira and Warren, 1980; Have, 2009). Instead, definite clause grammars use deduction rules from first-order predicate logic which are somewhat comparable to inverse rewrite functions and are commonly implemented in logic programming languages such as PROLOG and PRISM (Bratko, 1986; Sato and Kameya, 1997). The approach of PACFGs is in spirit also similar to adaptor grammars (Johnson et al., 2007a) which describe the same languages as standard context-free grammars and extend their probability model to increase the probability of frequently occurring derivation trees.

If no additional restrictions are put on the rules of ACFGs, then the class of languages generatable by ACFGs is the class of recursively enumerable languages, because all computation can be performed in unary rules. With the restriction of ACFGs in a Chomsky-normal form (see Section 6.1), languages beyond context-free complexity are still generatable. This expressivity is demonstrated by the following ACFG which generates the copy language $\{\boldsymbol{ww} \mid \boldsymbol{w} \in T^+\}$ over an alphabet $T = \{a, b\}$. The construction uses stacks of terminals as nonterminals and is as such similar to linear indexed grammars (Aho, 1968; Vijay-Shanker and Weir, 1993). The copy language is the language of the ACFG $G = (T, N, \text{Start}, R)$ with letters a and b as terminals, and two kinds of lists as nonterminals — one kind for generating a sequence $\boldsymbol{w} \in T^+$ and pushing its elements onto a stack and one kind for generating its copy by popping from the stack,

$$N = \{\textsc{MemWrite}(\boldsymbol{w}) \mid \boldsymbol{w} \in T^*\} \uplus \{\textsc{MemRead}(\boldsymbol{w}) \mid \boldsymbol{w} \in T^*\}. \tag{5.42}$$

The names $\textsc{MemWrite}(\boldsymbol{w})$ and $\textsc{MemRead}(\boldsymbol{w})$ are chosen to indicate that rewrite rules are only allowed to write (push) terminals to $\textsc{MemWrite}(\boldsymbol{w})$ and to read (pop) terminals from $\textsc{MemRead}(\boldsymbol{w})$. The grammar then essentially works in two phases. In the first phase, a sequence is generated and written to the memory. In the second phase, the sequence generated so far is again generated (copied) by reading from memory.

The start symbol of the grammar is the empty list onto which elements can be pushed, Start = $\textsc{MemWrite}(\varepsilon)$. The set of rules R consists of four partial functions:

$$\textsc{pusha}(A) = \begin{cases} \textsc{MemRead}(a) \quad \textsc{MemWrite}(\boldsymbol{w}a), & \text{if } A = \textsc{MemWrite}(\boldsymbol{w}) \\ \textsc{undefined}, & \text{otherwise} \end{cases} \tag{5.43}$$

$$\textsc{pushb}(A) = \begin{cases} \textsc{MemRead}(b) \quad \textsc{MemWrite}(\boldsymbol{w}b), & \text{if } A = \textsc{MemWrite}(\boldsymbol{w}) \\ \textsc{undefined}, & \text{otherwise} \end{cases} \tag{5.44}$$

$$\textsc{pop}(A) = \begin{cases} \textsc{MemRead}(\boldsymbol{w}_{1:|\boldsymbol{w}|-1}) \quad \textsc{MemRead}(\boldsymbol{w}_{|\boldsymbol{w}|}), & \text{if } \boldsymbol{w} \text{ stored in } A \text{ and } |\boldsymbol{w}| > 1 \\ \textsc{undefined}, & \text{otherwise} \end{cases} \tag{5.45}$$

$$\textsc{terminate}(A) = \begin{cases} \boldsymbol{w}_1, & \text{if } \boldsymbol{w} \text{ stored in } A \text{ and } |\boldsymbol{w}| = 1 \\ \textsc{undefined}, & \text{otherwise} \end{cases} \tag{5.46}$$

Here, the term "$\boldsymbol{w}$ stored in A" denotes the fact that either $A = \textsc{MemRead}(\boldsymbol{w})$ or $A = \textsc{MemWrite}(\boldsymbol{w})$. To illustrate the mechanics of this grammar, the leftmost derivation of the sequence $abbabb \in T^+$ is for example shown in Figure 5.5 The corresponding derivation tree is shown in Figure 5.6.

$$\begin{array}{lll}
\textsc{MemWrite}(\varepsilon) & \longrightarrow_{\textsc{pusha}} & \textsc{MemRead}(a)\ \textsc{MemWrite}(a) \\
 & \longrightarrow_{\textsc{terminate}} & a\ \textsc{MemWrite}(a) \\
 & \longrightarrow_{\textsc{pushb}} & a\ \textsc{MemRead}(b)\ \textsc{MemWrite}(ab) \\
 & \longrightarrow_{\textsc{terminate}} & a\ b\ \textsc{MemWrite}(ab) \\
 & \longrightarrow_{\textsc{pushb}} & a\ b\ \textsc{MemRead}(b)\ \textsc{MemWrite}(abb) \\
 & \longrightarrow_{\textsc{terminate}} & a\ b\ b\ \textsc{MemWrite}(abb) \\
 & \longrightarrow_{\textsc{pop}} & a\ b\ b\ \textsc{MemRead}(ab)\ \textsc{MemRead}(b) \\
 & \longrightarrow_{\textsc{pop}} & a\ b\ b\ \textsc{MemRead}(a)\ \textsc{MemRead}(b)\ \textsc{MemRead}(b) \\
 & \longrightarrow_{\textsc{terminate}} & a\ b\ b\ a\ \textsc{MemRead}(b)\ \textsc{MemRead}(b) \\
 & \longrightarrow_{\textsc{terminate}} & a\ b\ b\ a\ b\ \textsc{MemRead}(b) \\
 & \longrightarrow_{\textsc{terminate}} & a\ b\ b\ a\ b\ b
\end{array}$$

Figure 5.5 – Derivation of the sequence *abbabb* from the copy language.

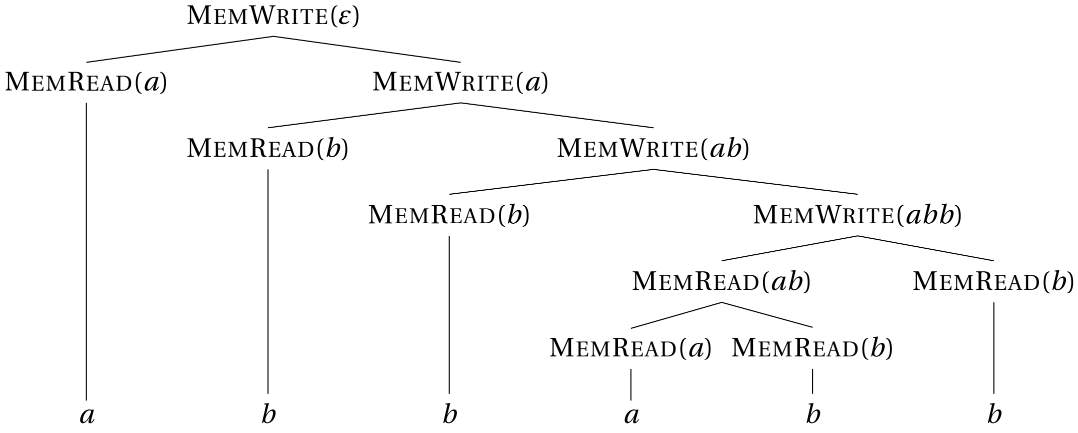


Figure 5.6 – Parse tree of the sequence *abbabb* from the copy language.

6 Semiring Parsing

The Cocke–Younger–Kasami (CYK) algorithm was originally formulated to answer the questions how many parse trees a sequence has according to a given standard context-free grammar, and what this parse forest looks like (Younger, 1967). The algorithm thereby solved the recognition problem for standard context-free grammars; that is, to find out whether a given sequence is generatable by a given grammar. Following research adopted the idea of the CYK algorithm to probabilistic context-free grammars of various forms (Earley, 1970), developing what is known as the inside-outside algorithm (Baker, 1979; Lari and Young, 1990; Stolcke, 1995). Modern parsing approaches generalize those algorithms to put them in the coherent algebraic framework of *semiring parsing* that decouples the parsing algorithm from the different quantities one might be interested about the parse forest (Goodman, 1998, 1999). The description and implementation of parsing algorithms can then be done in analogy to logic deduction systems — known as *parsing as deduction* (Shieber et al., 1995) — or in the framework of hypergraph parsing (Klein and Manning, 2004).

The essence of semiring parsing is to use a simple abstract representation to efficiently compute various quantities one might be interested about parse forests. In particular, this has the advantage that a parsing algorithm only needs to be implemented and tested once and can be reused to answer multiple queries.

This chapter starts by stating a probabilistic version of the CYK algorithm for abstract context-free grammars that calculates sequence probabilities. The algorithm is subsequently generalized to compute both the best derivation of a sequence and an efficient representation of the distribution over parse trees of a sequence. The representation of the parse-tree distribution as a mathematical object is novel and contributes to the research on parsing methods and grammar-learning algorithms. It is used later in the next chapter to learn a grammar's rewrite probabilities from sequential data. In particular, this chapter shows how to efficiently compute expected values of functions $g\colon R^* \to \mathbb{R}^d$ with respect to the distribution $p(\boldsymbol{r} \mid \boldsymbol{w})$ over derivations $\boldsymbol{r} \in \textsc{Der}(\boldsymbol{w})$ that yield an observed terminal sequence $\boldsymbol{w} \in T^+$,

$$\mathbb{E}_{p(\boldsymbol{r}|\boldsymbol{w})}\left[g(\boldsymbol{r})\right] = \sum_{\boldsymbol{r} \in \textsc{Der}(\boldsymbol{w})} p(\boldsymbol{r} \mid \boldsymbol{w}) g(\boldsymbol{r}), \tag{6.1}$$

where

$$p(\boldsymbol{r} \mid \boldsymbol{w}) \propto p(\boldsymbol{w} \mid \boldsymbol{r})\ p(\boldsymbol{r}) = \mathbb{1}(\boldsymbol{r}(\text{Start}) = \boldsymbol{w}))\, p(\boldsymbol{r}). \tag{6.2}$$

Here, the term $\mathbb{1}(\boldsymbol{r}(\text{Start}) = \boldsymbol{w})\, p(\boldsymbol{r})$ stands for the *indicator function* which maps a statement to its truth value:

$$\mathbb{1}(\boldsymbol{r}(\text{Start}) = \boldsymbol{w}) = \begin{cases} 1, & \text{if } \boldsymbol{r}(\text{Start}) = \boldsymbol{w} \\ 0, & \text{if } \boldsymbol{r}(\text{Start}) \neq \boldsymbol{w} \end{cases} \tag{6.3}$$

6.1 Parsing for PACFGs in Chomsky-normal form

The classic CYK algorithm assumes a grammar to be in *Chomsky normal-form.* That is, each rule $r \in R$ is either a unary rule which rewrites nonterminals into a terminals, $r : N \twoheadrightarrow T$, or a binary rule which rewrites nonterminals into sequences of two nonterminals, $r : N \twoheadrightarrow N^2$. Each standard context-free grammar can be converted in Chomsky-normal form in polynomial time by increasing the number of nonterminals and adjusting the rule set (Jurafsky and Martin, 2000). This section presents an adaptation of the CYK algorithm for abstract context-free grammars to calculate the probability $p_A(\boldsymbol{w}) = \sum_{\boldsymbol{r} \in \textsc{Der}(A,\boldsymbol{w})} p_A(\boldsymbol{r})$ that a non-empty sequence of terminals $\boldsymbol{w} \in T^+$ is generated from a nonterminal $A \in N$.

Let $G = (T, N, \text{Start}, R)$ be a PACFG in Chomsky-normal form. The CYK algorithm uses dynamic programming to calculate $p_A(\boldsymbol{w})$ via the following two recursive equations:

$$p_A(\boldsymbol{w}) = \sum_{\substack{r \in R_A \\ r(A) = \boldsymbol{w}}} p_A(r) \qquad \text{for } |\boldsymbol{w}| = 1 \tag{6.4}$$

$$p_A(\boldsymbol{w}) = \sum_{j=1}^{|\boldsymbol{w}|-1} \sum_{\substack{r \in R_A \\ \text{ar}(r)=2}} p_A(r)\ \ p_{r(A)_1}(\boldsymbol{w}_{1:j})\ \ p_{r(A)_2}(\boldsymbol{w}_{j+1:|\boldsymbol{w}|}) \qquad \text{for } |\boldsymbol{w}| \geq 2 \tag{6.5}$$

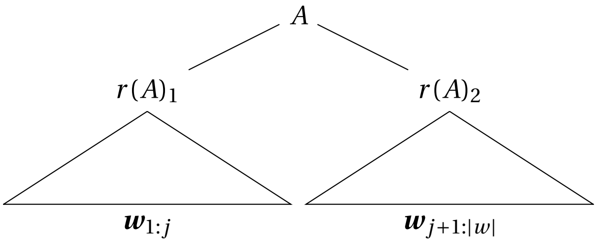


Figure 6.1 – Illustration of the computation of $p_A(\boldsymbol{w})$ in Equation 6.5.

The right-hand side of the second equation sums over the probabilities of all binary splits of $\boldsymbol{w}$; the computation is illustrated in Figure 6.1. The calculation is correct, because all derivation trees from a grammar in Chomsky-normal from are constructed either by exactly one unary termination rule or by an initial binary rule. The CYK algorithm uses the recursive equations to calculate $p_A(\boldsymbol{v})$ first for all of $\boldsymbol{w}$'s subsequences $\boldsymbol{v}$ of length 1, then of length 2, and so on. This bottom-up calculation can be performed efficiently, because the rewrite functions are bijective and thus invertible.

6.2 Semiring scores

The parsing algorithm presented in the last section is generalized in this section for using *scoring functions* instead of just probabilities. As it is shown later in this chapter, scoring functions can be used to perform various computations such as "Compute the most probable derivation of a given sequence" or "Sample a derivation which yields a given sequence".

A structural observation of the computation described by the Equations 6.4 and 6.5 reveals that the algorithm is based on the function

$$p\colon N \times R \to \mathbb{R}_{\geq 0}, \qquad (A, r) \mapsto p_A(r) \tag{6.6}$$

which assigns a probability to a pair of a nonterminal and a rule. Such pairs are also called *rule application* in the following. The computation moreover works for all scoring functions

$$\sigma\colon N \times R \to \mathbb{S}, \qquad (A, r) \mapsto \sigma_A(r). \tag{6.7}$$

that map every rule application (A, r) to a score $\sigma_A(r)$ in a space $\mathbb{S}$ which has a sensible notion of addition and multiplication, such that

$$\sigma_A(r) \neq 0 \iff A \in \operatorname{dom}(r). \tag{6.8}$$

Equations 6.4 and 6.5 are then generalized to scoring functions by:

$$\sigma_A(\boldsymbol{w}) = \sum_{\substack{r \in R_A \\ r(A) = \boldsymbol{w}}} \sigma_A(r) \qquad \text{for } |\boldsymbol{w}| = 1 \tag{6.9}$$

$$\sigma_A(\boldsymbol{w}) = \sum_{j=1}^{|\boldsymbol{w}|-1} \sum_{\substack{r \in R_A \\ \operatorname{ar}(r) = 2}} \sigma_A(r)\ \sigma_{r(A)_1}(\boldsymbol{w}_{1:j})\ \sigma_{r(A)_2}(\boldsymbol{w}_{j+1:|\boldsymbol{w}|}) \qquad \text{for } |\boldsymbol{w}| \geq 2 \tag{6.10}$$

Analogous to probability scores, the continuation of the function $\sigma_A\colon R \to \mathbb{S}$ from rules to derivations is given by

$$\sigma_A(\boldsymbol{r}) = \prod_{k=1}^{|r|} \sigma_{\boldsymbol{r}_{1:k-1}(A)}(\boldsymbol{r}_k) \tag{6.11}$$

for derivations $\boldsymbol{r} \in R^*$. The score of a sequence of terminals $\boldsymbol{w} \in T^+$ is defined as

$$\sigma_A(\boldsymbol{w}) = \sum_{\boldsymbol{r} \in \mathrm{DER}(A, \boldsymbol{w})} \sigma_A(\boldsymbol{r}). \tag{6.12}$$

The precise meaning of a "sensible notion of addition and multiplication" will be defined later by referring to the well-known theory of semirings. First, however, a simple example of a scoring that is different from the probability scoring considered above is described.

As a first simple example, consider the scoring function that maps a nonterminal $A \in N$ and a rule $r \in R$ to the natural number 1 if r is applicable to A and to 0 otherwise,

$$\sigma\colon N \times R \to \mathbb{N}, \qquad \sigma_A(r) = \mathbb{1}(A \in \mathrm{dom}(r)). \tag{6.13}$$

For this scoring function, the Equations 6.9 and 6.10 simplify to:

$$\sigma_A(\boldsymbol{w}) = \sum_{\substack{r \in R_A \\ r(A) = \boldsymbol{w}}} \mathbb{1}(A \in \mathrm{dom}(r)) = \sum_{\substack{r \in R_A \\ r(A) = \boldsymbol{w}}} 1 = |\mathrm{DER}(A, \boldsymbol{w})| \qquad \text{for } |\boldsymbol{w}| = 1 \tag{6.14}$$

$$\sigma_A(\boldsymbol{w}) = \sum_{j=1}^{|\boldsymbol{w}|-1} \sum_{\substack{r \in R_A \\ \mathrm{ar}(r) = 2}} \underbrace{\sigma_{r(A)_1}(\boldsymbol{w}_{1:j})}_{\substack{\text{\# derivations} \\ \text{left child}}} \; \underbrace{\sigma_{r(A)_2}(\boldsymbol{w}_{j+1:|\boldsymbol{w}|})}_{\substack{\text{\# derivations} \\ \text{right child}}} \qquad \text{for } |\boldsymbol{w}| \geq 2 \tag{6.15}$$

Therefore, it is easily proven by induction over the length of $\boldsymbol{w}$ that for this scoring function, $\sigma_A(\boldsymbol{w})$ computes the number of derivations of $\boldsymbol{w}$ from A: For the base case $|\boldsymbol{w}| = 1$, this follows directly from Equation 6.14. For the induction step, let n be a natural number such that $\sigma_A(\boldsymbol{w}) = |\mathrm{DER}(A, \boldsymbol{w})|$ for all $\boldsymbol{w} \in T^+$ with $|\boldsymbol{w}| \leq n$ and let $\boldsymbol{v} \in T^{n+1}$ and $j \in \{1, \ldots, n\}$. Since $|\boldsymbol{v}_{1:j}| \leq n$ and $|\boldsymbol{v}_{j+1:n+1}| \leq n$, $\sigma_B(\boldsymbol{v}_{1:j}) = |\mathrm{DER}(B, \boldsymbol{v}_{1:j})|$ and $\sigma_B(\boldsymbol{v}_{j+1:n+1}) = |\mathrm{DER}(B, \boldsymbol{v}_{j+1:n+1})|$ for all nonterminals $B \in N$ by the induction hypothesis. Since the grammar is assumed to be in Chomsky-normal form and $|\boldsymbol{v}| > 1$, all derivations of $\boldsymbol{v}$ from A must start with a binary rule. Therefore, Equation 6.15 completes the proof.

The above example can be generalized to establish an intuition for the interpretation of addition and multiplication in parsing. Addition stands for a choice from two options. In the counting example, addition is the usual addition, because the interest is to calculate the total number of all possible derivations. Multiplication stands for the combination of possibilities. In Equation 6.15, the number of derivations of the left child of A, denoted by $r(A)_1$, and the number of derivations of the right child of $r(A)_1$ are multiplied, because each combination can be used to construct a valid derivation of $\boldsymbol{w}$ from A. These two intuitions will be also helpful to understand the more complex semirings described below.

Semiring A *semiring* $\mathbb{S} = (S, +, \cdot, 0, 1)$ consists of two monoids $(S, +, 0)$ and $(S, \cdot, 1)$ (see Section 5.2 for the definition of a monoid) such that for all $s, t, u \in S$:

$$s + t = t + s \qquad \text{(commutativity of addition)} \tag{6.16}$$

$$s \cdot 0 = 0 = 0 \cdot s \qquad \text{(absorption by zero)} \tag{6.17}$$

$$u \cdot (s + t) = u \cdot s + u \cdot t \qquad \text{(left distributivity)} \tag{6.18}$$

$$(s + t) \cdot u = s \cdot u + t \cdot u \qquad \text{(right distributivity)} \tag{6.19}$$

The set S is called the *carrier set* of the semiring. Usually, $s \cdot t$ is abbreviated by st like it is the case with the usual multiplication.

The semirings of the probability scoring and the count scoring are $(\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$ and $(\mathbb{N}, +, \cdot, 0, 1)$, respectively, both with the usual addition and multiplication. The carrier set of the semiring of the probability scoring is the set of all non-negative real numbers, to ensure that the addition of two numbers always stays in the semiring. During parsing with a probabilistic grammar, numbers greater than one are, however, never reached.

A more advanced example is the semiring $\textsc{Best}_R$ of the scoring function

$$\sigma\colon N \times R \to \textsc{Best}_R, \qquad \sigma_A(r) = (r, p_A(r)) \tag{6.20}$$

for which $\sigma_A(\boldsymbol{w})$ is the pair of the most probable derivation of $\boldsymbol{w}$ from A and its probability. The semiring $\textsc{Best}_R$ is defined by:

$$\textsc{Best}_R = ((R^* \times \mathbb{R}_{\geq 0}), +, \cdot, (\varepsilon, 0), (\varepsilon, 1)) \tag{6.21}$$

$$(\boldsymbol{r}, q) + (\boldsymbol{r}', q') = \begin{cases} (\boldsymbol{r}, q), & \text{if } q > q' \text{ or } (q = q' \text{ and } \boldsymbol{r} > \boldsymbol{r}') \\ (\boldsymbol{r}', q'), & \text{otherwise} \end{cases} \tag{6.22}$$

$$(\boldsymbol{r}, q) \cdot (\boldsymbol{r}', q') = (\boldsymbol{r}\boldsymbol{r}', qq') \tag{6.23}$$

where $\boldsymbol{r} > \boldsymbol{r}'$ denotes that $\boldsymbol{r}$ is lexicographically greater than $\boldsymbol{r}'$ with respect to any arbitrarily chosen total order on the finite set of rules R. This technicality is needed to satisfy the commutativity of the semiring's addition. The addition of this semiring selects the best of two derivations, and the multiplication concatenates derivations and multiplies their probabilities. Therefore, the zero element must have probability zero and the identity element must have probability one.

The semiring $\textsc{Best}_R$ is an example of a *non-commutative* semiring. That is, a semiring in which the multiplication is not commutative. The reason why the multiplication of $\textsc{Best}_R$ is not commutative is that in Equation 6.23, the derivations $\boldsymbol{r}$ and $\boldsymbol{r}'$, which are represented as lists of rules, are concatenated and the concatenation of lists is not commutative. It is important to acknowledge that not all useful semirings are commutative, because it implies that the order of the factors in Equation 6.10 matters.

To summarize, the generalization of the CYK algorithm (also known as the inside algorithm) to arbitrary semiring scoring functions enables the usage of one generic parsing algorithm for various queries about the parse forest such as the total probability or the most probable tree of the parse forest. The semiring theory combines the explicit mathematical formalization of these query calculations with a concise computational implementation. This theory is even more powerful. The following sections first generalize the CYK algorithm to a broader class of PACFGs and then present a scoring function that calculates a compact representation of the distribution of a sequence's derivations. More examples of what can be calculated in the semiring-parsing framework are, the number of parse trees, the probability of the best parse tree, the list of all parse trees, and the list of the best k parse trees for any number k (Goodman, 1999).

6.3 Parsing for PACFGs with unary rules

Grammars in Chomsky-normal from are modeling tools with the unnecessary limitation that only binary-branching rules and unary terminal rules are allowed. This section shows how the semiring version of the inside algorithm described in the last section is extended to grammars that allow arbitrary unary and binary loops. Unary rules can, for instance, be used to emulate multiple start symbols or to model reinterpretations of chord symbols. An important example of reinterpretation in the context of musical grammar is modulation as formalized by Rohrmeier (2011) and implemented by Harasim et al. (2018). In that formalization, scale degrees can be reinterpreted as a first scale degree in the corresponding key. The scale degree IV in they of C major can, for example, be reinterpreted as scale degree I in the key of F major. On one hand, such a reinterpretation is necessarily a cognitive computation which is beneficial to be includeable in a cognitively motivated grammar framework, on the other hand, reinterpretations introduce ambiguities that might be difficult to handle in practice (Harasim et al., 2018).

As mentioned above, this section proposes a generalization of the Chomsky-normal form which permits arbitrary rules of arity 1 and 2. Rules of arity 3 and higher must be split into multiple rules, similar to the conversion of standard context-free grammars into Chomsky-normal form. Consider for example a rule $r\colon N ⇸ N^3$ of arity 3 and let $A \in \mathrm{dom}(r)$. This rule can be split into two rules $r'\colon N ⇸ N \times X$ and $r''\colon X \to N \times N$ such that $r'(A) = r(A)_1\ X_A$ and $r''(X_A) = r(A)_2\ r(A)_3$, where $X = \{X_A \mid A \in \mathrm{dom}(r)\}$ is a new set of additional nonterminals. This conversion can be performed automatically if the set of nonterminals is finite. If otherwise the grammar comprises infinitely many nonterminals, then there is no automatic conversion known that works for all abstract context-free grammars. For an overview over normal forms of standard context-free grammars and their application to parsing see for example Lange and Leiß (2009).

Parsing is more complicated for grammars that allow arbitrary unary rules than for grammars in Chomsky-normal form. The reason is that it might be possible that some rules can be applied in a loop. This happens for example if two distinct nonterminals A and B are derivable from each other (e.g., $\textsc{Der}(A,B) \neq \emptyset \neq \textsc{Der}(B,A)$). In that case, there are infinitely many derivations of A from B and vice versa. To control that only finitely many of such loops occur, a notation is introduced that describes the set of nonterminals A' that are traversed by unary derivations (e.g., sequences of unary rules) of a nonterminal $B \in N$ from a nonterminal $A \in N$:

$$\textsc{Trav}(A,B) = \{A' \in N \mid \exists \boldsymbol{r}, \boldsymbol{r}' \in R^* : r(A) = A' \text{ and } r'(A') = B\} \tag{6.24}$$

Note that since $\boldsymbol{r}$ or $\boldsymbol{r}'$ can be chosen as the empty derivation, $A, B \in \textsc{Trav}(A,B)$ for all $A, B \in N$.

1-2-normal form An abstract context-free grammar is in *1-2-normal form* if all rules have arity 1 or 2, $\{\mathrm{ar}(r) \mid r \in R\} \subseteq \{1,2\}$, and if $\textsc{Trav}(A,B)$ is finite for all $A, B \in N$.

The main difference between the Chomsky-normal form and its proposed generalization is that the former forbids unary derivations of nonterminals (e.g., $\textsc{Der}(A,B) = \emptyset$ for all $A, B \in N$) while the latter allows infinitely many unary derivations and therefore infinitely many derivations of a sequence of terminals. The finite cardinality of $\textsc{Trav}(A,B)$, however, restricts the kind of infinity that is allowed. It implies that the set of derivations $\textsc{Der}(A,B)$ can only be infinitely large if there is a nonterminal $A' \in \textsc{Trav}(A,B)$ which has a nontrivial derivation from itself. That is, there exists a derivation $\boldsymbol{r} \in \textsc{Der}(A', A')$ such that $|\boldsymbol{r}| > 0$. Such a derivation is called a *unary loop* at the nonterminal A'. A simple academic example of a standard context-free grammar that has a unary loop at the start symbol is

$$G = (\{a\}, \{\text{Start}, A\}, \text{Start}, \{\text{Start} \longrightarrow A, A \longrightarrow \text{Start}, A \longrightarrow a\}). \tag{6.25}$$

This grammar is in 1-2-normal form, but not in Chomsky-normal form. In fact, all ACFGs with rules of arity 1 and 2 are in 1-2-normal form if the set of nonterminals is finite, which is in particular the case for standard context-free grammars. Also, each ACFG in Chomsky-normal form is in 1-2-normal form.

Let now $G = (T, N, \text{Start}, R)$ be an abstract context-free grammar in 1-2-normal form and let $\sigma_{(-)}(=) : N \times R \to \mathbb{S}$ be a scoring function. For all nonterminals $A \in N$ and terminal sequences $\boldsymbol{w} \in T^*$, the score $\sigma_A(\boldsymbol{w})$ can then be calculated using the following recursive equations

$$\begin{aligned} \sigma_A(\boldsymbol{w}) &= \sum_{\boldsymbol{r} \in \textsc{Der}(A,\boldsymbol{w})} \sigma_A(\boldsymbol{r}) && \text{for } |\boldsymbol{w}| = 1 \\ \sigma_A(\boldsymbol{w}) &= \sum_{B \in N} \underbrace{\left(\sum_{\boldsymbol{r} \in \textsc{Der}(A,B)} \sigma_A(\boldsymbol{r}) \right)}_{\text{unary derivations}} \underbrace{\sum_{j=1}^{|\boldsymbol{w}|-1} \sum_{\substack{r \in R_B \\ \mathrm{ar}(r)=2}} \sigma_B(r)\ \sigma_{r(B)_1}(\boldsymbol{w}_{1:j})\ \sigma_{r(B)_2}(\boldsymbol{w}_{j+1:|\boldsymbol{w}|})}_{\text{binary splits}} && \text{for } |\boldsymbol{w}| \geq 2 \end{aligned} \tag{6.26}$$

where $\sigma_{r(B)_i}(\boldsymbol{v}) = \mathbb{1}(r(B)_i = \boldsymbol{v})$ if $r(B)_i$ is a terminal ($r(B)_i \in T$). These equations are a natural extension of the Equations 6.9 and 6.10 with unary derivations; The term that sums over the

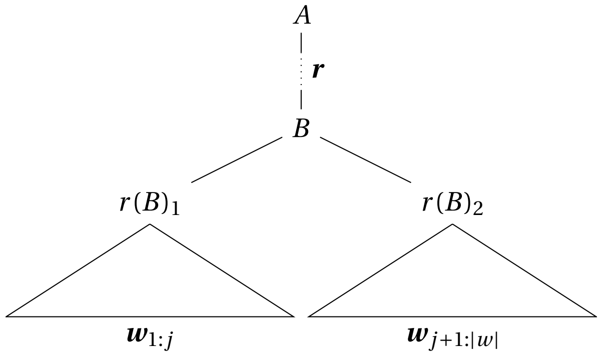


Figure 6.2 – Illustration of the computation of $p_A(\boldsymbol{w})$ in Equation 6.26.

binary splits is equivalent to the right-hand side of equation 6.10. The extended calculation is illustrated in Figure 6.2. To calculate the scores $\sum_{\boldsymbol{r} \in \mathrm{DER}(A,B)} \sigma_A(\boldsymbol{r})$ of infinite sets of unary derivations, the concept of a closed semiring is used.

Closed semiring A *closed semiring* (also called *star semiring*) is an algebraic structure $(S, +, \cdot, ^*, 0, 1)$ for which $(S, +, \cdot, 0, 1)$ is a semiring and $(-)^* \colon S \to S,\ s \mapsto s^*$ is a unary operation that satisfies the equalities

$$s^* = 1 + s \cdot s^* = 1 + s^* \cdot s \tag{6.27}$$

for all $s \in S$. The value s^* is called the *closure* of s, and $(-)^*$ is called the closure operation.

The intuition behind the closure operation is that

$$s^* = 1 + ss^* = 1 + s(1 + ss^*) = 1 + s + sss^* = 1 + s + ss + sss + \ldots \tag{6.28}$$

formalizes the geometric series in a closed semiring. For parsing with a PACFG, addition and multiplication are used to express choice and concatenation of derivations, respectively. The closure operation expresses iterated application of, for example, unary rules. The strength of closed semirings is their wide applicability; A closure operation can be defined for important semirings (as shown in the next section) which do not support the calculation of arbitrary infinite sums. The usage of closed semirings constitutes a slight generalization to the semiring-parsing framework proposed by Goodman (1999) who assumes semirings to be *complete*, that is all infinite sums are always calculable within the semiring.

Using the usual addition and multiplication of numbers, the semirings $(\mathbb{R}_{\geq 0} \cup \{\infty\}, +, \cdot, 0, 1)$ and $(\mathbb{N} \cup \{\infty\}, +, \cdot, 0, 1)$ form closed semirings with $x^* = (1-x)^{-1}$ for $x < 1$ and $x^* = \infty$ for $x \geq 1$. In this case, the closure of an element is an actual geometric series. The semiring

$$\mathrm{BEST}_R = ((R^* \times \mathbb{R}_{\geq 0}), +, \cdot, (\varepsilon, 0), (\varepsilon, 1)) \tag{6.29}$$

can be extended with an element (ε, ∞) to form a closed semiring with

$$(\boldsymbol{r}, q)^* = \begin{cases} (\varepsilon, 1), & \text{if } q < 1 \\ (\varepsilon, \infty), & \text{otherwise.} \end{cases} \tag{6.30}$$

The second case does, however, not occur during parsing with probabilistic context-free grammars. The first case reflects in particular that the most probable derivation never has unary loops.

In the following paragraphs, the sum $\sum_{\boldsymbol{r} \in \mathrm{D\scriptstyle ER}(A,B)} \sigma_A(\boldsymbol{r})$ is calculated by the usage of matrices over closed semirings. This sum is the only term whose computation was left unexplained so far in Equation 6.10. The calculation is efficiently performable since for every closed semiring $\mathbb{S}$ and positive natural number $n \in \mathbb{N}_{>0}$, the square matrices $\mathbb{S}^{n \times n}$ form again a closed semiring with the usual addition and multiplication of matrices (based on the operations of $\mathbb{S}$). The closure $\boldsymbol{M}^*$ of a matrix $\boldsymbol{M}$ can be calculated recursively via an equation for block matrices

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix} \tag{6.31}$$

where $\boldsymbol{A} \in \mathbb{S}^{n' \times n'}$, $\boldsymbol{B} \in \mathbb{S}^{n-n' \times n'}$, $\boldsymbol{C} \in \mathbb{S}^{n' \times n-n'}$, and $\boldsymbol{D} \in \mathbb{S}^{n-n' \times n-n'}$, for some $0 < n' < n$. Lehmann (1977) showed that $\boldsymbol{M}$ satisfies the equality

$$\boldsymbol{M}^* = \begin{pmatrix} \boldsymbol{A}^* + \boldsymbol{B}'\boldsymbol{E}^*\boldsymbol{C}' & \boldsymbol{B}'\boldsymbol{E}^* \\ \boldsymbol{E}^*\boldsymbol{C}' & \boldsymbol{E}^* \end{pmatrix} \tag{6.32}$$

where $\boldsymbol{B}' = \boldsymbol{A}^*\boldsymbol{B}$, $\boldsymbol{C}' = \boldsymbol{C}\boldsymbol{A}'$, and $\boldsymbol{E} = \boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}^*\boldsymbol{B}$. The algorithm which calculates the closure of a matrix $\boldsymbol{M}$ splits it into the blocks $\boldsymbol{A} = \boldsymbol{M}_{1,1} \in \mathbb{S}^{1\times 1}$, $\boldsymbol{B} = \boldsymbol{M}_{2:n,1} \in \mathbb{S}^{n-1\times 1}$, $\boldsymbol{C} = \boldsymbol{M}_{1,2:n} \in \mathbb{S}^{1\times n-1}$, and $\boldsymbol{D} = \boldsymbol{M}_{2:n,2:n} \in \mathbb{S}^{n-1\times n-1}$ and uses the fact that the closure of a singleton matrix reduces to the closure of its single element, $\boldsymbol{A}^* = a^*$ for some $a \in \mathbb{S}$. The underlying calculations of this algorithm are similar to the Floyd-Warshall algorithm for all-pairs shortest paths (Dolan, 2013; Floyd, 1962)

To calculate $\sum_{\boldsymbol{r} \in \mathrm{D\scriptstyle ER}(A,B)} \sigma_A(\boldsymbol{r})$ for $A \in N$ and $B \in T \uplus N$, set $n = |\mathrm{T\scriptstyle RAV}(A,B)| \in \mathbb{N}$ to the finite number of traversed nonterminals and define the score matrix $\boldsymbol{M} \in \mathbb{S}^{n\times n}$ of single unary transitions as

$$\boldsymbol{M}_{C,D} = \sum_{\substack{r \in R_C \\ r(C)=D}} \sigma_C(r) \tag{6.33}$$

for nonterminals $C, D \in \mathrm{T\scriptstyle RAV}(A,B)$. To simplify the notation, the arbitrary bijection that converts nonterminals from $\mathrm{T\scriptstyle RAV}(A,B)$ into natural numbers from 1 to n is omitted.

For all $C, D \in \mathrm{T\scriptstyle RAV}(A,B)$, the value $\boldsymbol{M}_{C,D}$ is the sum of the scores of all unary rules that rewrite C into D. The k-th power of $\boldsymbol{M}$ contains at the position C, D the summed scores of all derivations of D from C with length k (Dolan, 2013),

$$\boldsymbol{M}^k_{C,D} = \sum_{\substack{\boldsymbol{r}\in\textsc{Der}(C,D)\\|\boldsymbol{r}|=k}} \sigma_C(\boldsymbol{r}). \tag{6.34}$$

The interpretation of the closure $\boldsymbol{M}^*$ as a geometric series therefore computes the searched value,

$$\sum_{\boldsymbol{r}\in\textsc{Der}(A,B)} \sigma_A(\boldsymbol{r}) = \boldsymbol{M}^*_{A,B}. \tag{6.35}$$

6.4 The semiring of derivation distributions

Semiring parsing cannot only be used to calculate probabilities or best derivations, but also to compactly represent the full distribution of derivations for a given sequence of terminals $\boldsymbol{w}$. To the best of our knowledge, this is the first proposal for representing a distribution of parse trees as a single mathematical object in the semiring parsing framework. From this representation, derivations $\boldsymbol{r}^1,\ldots,\boldsymbol{r}^n \in \textsc{Der}(\boldsymbol{w})$ can be sampled efficiently to approximate expected values

$$\mathbb{E}_{p(\boldsymbol{r}|\boldsymbol{w})}\left[g(\boldsymbol{r})\right] \approx \frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{r}^i) \tag{6.36}$$

for functions $g\colon R^* \to \mathbb{R}^d$ ($d \in \mathbb{N}_+$). This stochastic approximation is called the *Monte-Carlo estimate* of the expected value. It is used in the next chapter to learn the parameters of a probabilistic grammar from sequential data. The more traditional calculation method for the expected value $\mathbb{E}_{p(\boldsymbol{r}|\boldsymbol{w})}\left[g(\boldsymbol{r})\right]$ is the inside-outside algorithm (Baker, 1979; Lari and Young, 1990). That algorithm was originally formulated for grammars in Chomsky-normal form and its generalization to arbitrary unary rules is not straight-forward (Stolcke, 1995). The Monte-Carlo estimate is used in this study, because of its simplicity and easy generalizability to grammars of various forms.

The values of the closed semiring $\textsc{Dist}_R$ of distributions over derivations $\boldsymbol{r} \in R^*$ are based on abstract syntax trees enriched with probabilities. The carrier set of $\textsc{Dist}_R$ is based on a set D which is recursively defined by

$$\begin{aligned} D = \{&\textsc{Zero},\ \textsc{One},\ \textsc{Val}(r,q),\ \textsc{Add}(d,d',q),\ \textsc{Mul}(d,d',q),\ \textsc{Star}(d,q) \\ &\mid r \in R,\ d,d' \in D,\ q \in \mathbb{R}_{\geq 0} \uplus \{\infty\}\}. \end{aligned} \tag{6.37}$$

Thereby, Zero, One, Val, Add, Mul, and Star are best understood as tags of the abstract syntax tree without semantic meaning (they are also interpretable as injective value constructors).

The operations are based on

$$0 = \textsc{Zero} \tag{6.38}$$

$$1 = \textsc{One} \tag{6.39}$$

$$d + d' = \textsc{Add}(d, d', \textsc{prob}_D(d) + \textsc{prob}_D(d')) \tag{6.40}$$

$$d \cdot d' = \textsc{Mul}(d, d', \textsc{prob}_D(d) \cdot \textsc{prob}_D(d')) \tag{6.41}$$

$$d^* = \textsc{Star}(d, \textsc{prob}_D(d)^*) \tag{6.42}$$

for $d, d' \in D$ and the probability-selecting function $\textsc{prob}_D \colon D \to \mathbb{R}_{\geq 0} \uplus \{\infty\}$ with:

$$\textsc{prob}_D(\textsc{Zero}) = 0 \tag{6.43}$$

$$\textsc{prob}_D(\textsc{One}) = 1 \tag{6.44}$$

$$\textsc{prob}_D(\textsc{Val}(r, q)) = q \tag{6.45}$$

$$\textsc{prob}_D(\textsc{Add}(d, d', q)) = q \tag{6.46}$$

$$\textsc{prob}_D(\textsc{Mul}(d, d', q)) = q \tag{6.47}$$

$$\textsc{prob}_D(\textsc{Star}(d, q)) = q \tag{6.48}$$

The set D and the operations defined on it are not yet a semiring, because the defining equations (i.e., the semiring axioms; e.g., distributivity) are not yet satisfied. The carrier set of $\textsc{Dist}_R$ is therefore defined as coarsest partition on D for which all required equalities of a closed semiring hold. The partition (i.e., the set of equivalence classes) is denoted by $[D]$ and the equivalence class of any $d \in D$ is denoted by $[d]$. This construction is analogous to the standard construction of free algebras (Ihringer and Gumm, 2003; Burris and Sankappanavar, 2012). For example, let $r \in R$ be a rule, let $A \in \mathrm{dom}(r)$ be a nonterminal in r's domain, and let $q := p_A(r)$. Then $\textsc{Val}(r, q)$ and $\textsc{Add}(\textsc{Val}(r, q), \textsc{Zero})$ are in the same equivalence class because $\textsc{Zero}$ is the neutral element of the addition, $\textsc{Add}(\textsc{Val}(r, q), \textsc{Zero}) \in [\textsc{Val}(r, q)]$. This equivalence also contains element such as $\textsc{Add}(\textsc{Zero}\textsc{Val}(r, q))$ as $\textsc{Mul}(\textsc{Val}(r, q), \textsc{One})$. All these values are distinct in D, but identified in $[D]$. Furthermore, all equivalence classes are infinitely large.

It is easy to show by induction over the definition of D (Equation 6.37) that all elements of an equivalence class map to the same value under the probability-selecting function $\textsc{prob}_D$,

$$\forall d, d' \in D\colon \quad [d] = [d'] \implies \textsc{prob}_D(d) = \textsc{prob}_D(d'). \tag{6.49}$$

The probability function on $[D]$ can therefore be defined on representatives of equivalence classes,

$$\textsc{prob} \colon [D] \to \mathbb{R}_{\geq 0} \uplus \{\infty\}, \qquad \textsc{prob}([d]) = \textsc{prob}_D(d). \tag{6.50}$$

Algorithm 2 Sample derivation from an element of $\textsc{Dist}_R$

```
Input: $[d] \in \textsc{Dist}_R$ such that $0 < \textsc{prob}([d]) < \infty$
Output: derivation $\boldsymbol{r} \in R^*$

function SAMPLE_DERIVATION(d)
    if d match ONE then
        return $\varepsilon$                                    ▷ empty derivation
    else if d match VAL$(r, q)$ then
        return $r$                                              ▷ singleton derivation
    else if d match ADD$(d', d'', q)$ then
        go_left ← sample from Bernoulli(PROB$(d')/q$)
        if go_left then
            SAMPLE_DERIVATION$(d')$
        else
            SAMPLE_DERIVATION$(d'')$
        end if
    else if d match MUL$(d', d'', q)$ then
        $\boldsymbol{r}'$ ← SAMPLE_DERIVATION$(d')$
        $\boldsymbol{r}''$ ← SAMPLE_DERIVATION$(d'')$
        return $\boldsymbol{r}'\boldsymbol{r}''$                ▷ sequence concatenation
    else if d match STAR$(d', q)$ then
        iterate ← sample from Bernoulli(PROB$(d')$)
        if iterate then
            $\boldsymbol{r}'$ ← SAMPLE_DERIVATION$(d')$      ▷ sample from distribution which is iterated
            $\boldsymbol{r}''$ ← SAMPLE_DERIVATION$(d)$                             ▷ iterate sampling
            return $\boldsymbol{r}'\boldsymbol{r}''$                           ▷ sequence concatenation
        else
            return $\varepsilon$                                       ▷ empty derivation
        end if
    end match
end function
```

The closed semiring $\textsc{Dist}_R$ of distributions over derivations is now given by

$$\textsc{Dist}_R = ([D], +, \cdot, ^*, [\textsc{Zero}], [\textsc{One}]) \tag{6.51}$$

where all operations are defined on representatives ($d, d' \in D$):

$$[d] + [d'] = [d + d'] \tag{6.52}$$

$$[d] \cdot [d'] = [d \cdot d'] \tag{6.53}$$

$$[d]^* = [d^*] \tag{6.54}$$

The respective scoring function is

$$\sigma\colon N \times R \to \textsc{Dist}_R, \qquad \sigma_A(r) = [\textsc{Val}(r, p_A(r))]. \tag{6.55}$$

Note that because all operations are defined on representatives, implementations of $\textsc{Dist}_R$ can simply work with the elements of D. From all $d \in D$ with $0 < \textsc{prob}_D(d) < \infty$, derivations $\boldsymbol{r} \in R^*$ can be sampled as described by the recursive procedure shown in Algorithm 2. The samples can be used to approximate expected values as shown in Equation 6.36.

6.5 Efficient parsing of product grammars

The definition of product grammars given in Section 5.5 is precise and simple, but not efficient for parsing. A naive approach to parsing against a product grammar with finitely many symbols would enumerate all product nonterminals (e.g., the full Cartesian product) and memoize the inverted rewrite rules on these nonterminals. This section shows how the inefficient quadratic blow-up in the number of nonterminals can be avoided for grammars in Chomsky-normal form. The main idea is thereby to generalize the independence assumption of Equation 5.38,

$$p_{(A,A')}((r, r')) \propto p_A(r)\; p_{A'}(r') \tag{6.56}$$

for product rules (r, r') and product nonterminals (A, A').

Let $G \bowtie G'$ be the product of the grammars $G = (T, N, \text{Start}, R)$ and $G' = (T', N', \text{Start}', R')$ with scoring functions $\sigma\colon N \times R \to \mathbb{S}$ and $\sigma'\colon N' \times R' \to \mathbb{S}'$, respectively. Since σ and σ' generally map into different semirings, an operation $\diamond$ is assumed that combines scores $\sigma_A(r)$ and $\sigma_{A'}(r')$ into a semiring $\mathbb{S}''$. Using this operation, the product scoring function is defined by

$$\sigma \bowtie \sigma'\colon (N \times N') \times (R \bowtie R') \to \mathbb{S}'', \qquad \sigma \bowtie \sigma'_{(A,A')}((r, r')) = \sigma_A(r) \diamond \sigma'_{A'}(r'). \tag{6.57}$$

In the example where both σ and σ' yield the bare rewrite probabilities, the operation $\diamond$ can be chosen as the usual product followed by a renormalization. In the example where $\sigma\colon N \times R \to \textsc{Best}_R$ and $\sigma'\colon N' \times R' \to \textsc{Best}_{R'}$, the combining operation which pairs the component rules and calculates their product-rule probability can be chosen,

$$\sigma_A(r) \diamond \sigma'_{A'}(r') = (r, p_A(r)) \diamond (r', p_{A'}(r')) = ((r, r'), p_{(A,A')}((r, r'))). \tag{6.58}$$

If $\mathrm{ar}(r) = \mathrm{ar}(r')$, then $((r, r'), p_{(A,A')}((r, r'))) \in \textsc{Best}_{R \bowtie R'}$. The condition of equal arity will always be met in the proposed parsing computation below.

The generic Equations 6.9 and 6.10 specialize for the scoring function $\sigma \bowtie \sigma'$ of the product grammar to

$$\sigma \bowtie \sigma'_{(A,A')}(\boldsymbol{w}) = \sum_{\substack{(r,r') \in R_A \bowtie R_{A'} \\ r(A) = \boldsymbol{v} \\ r'(A') = \boldsymbol{v}'}} \sigma_A(r) \diamond \sigma'_{A'}(r') \qquad \text{for } |\boldsymbol{w}| = 1 \tag{6.59}$$

$$\sigma \bowtie \sigma'_{(A,A')}(\boldsymbol{w}) = \sum_{j=1}^{|\boldsymbol{w}|-1} \sum_{\substack{r \in R_A \\ \mathrm{ar}(r)=2}} \sum_{\substack{r \in R_{A'} \\ \mathrm{ar}(r')=2}} (\sigma_A(r) \diamond \sigma'_{A'}(r')) \qquad \text{for } |\boldsymbol{w}| \geq 2 \tag{6.60}$$

$$\sigma \bowtie \sigma'_{(r(A)_1, r'(A')_1)}(\boldsymbol{w}_{1:j}) \;\; \sigma \bowtie \sigma'_{(r(A)_2, r'(A')_2)}(\boldsymbol{w}_{j+1:|\boldsymbol{w}|})$$

where $\boldsymbol{w} = \textsc{zip}(\boldsymbol{v}, \boldsymbol{v}')$ (i.e., $\boldsymbol{w}_j = (\boldsymbol{v}_j, \boldsymbol{v}'_j)$). It is therefore sufficient to parse the component grammars individually at each step. In other words, the combined grammar is computed on-the-fly to achieve efficiency.

7 Bayesian Inference

The core idea of Bayesian statistics is to consider unknown model parameters as random variables. The distribution of the parameters then quantifies the plausibility of a parameter instantiation. For a generative grammar model, this implies that the rewrite probabilities are considered as random variables. Depending on the concrete grammar model, the rewrite probabilities are either directly modeled as parameters — such as for probabilistic standard context-free grammars — or are calculated from more sophisticated parameterizations — as proposed in this chapter.

This chapter first proposes a finite dimensional parameterization for PACFGs. Subsequently, a prior distribution over the parameter space with good mathematical properties is presented. Then Coordinate Ascent Variational Inference (CAVI) is described and applied to derive iterative update equations for the inference of a PACFG's parameters. Finally, the last section summarizes the proposed model and the inference procedure of its parameters from sequential data.

The models and algorithms constructed in this chapter are generalizations of the variational Bayesian inference procedure for probabilistic standard context-free grammars by Kurihara and Sato (2004). The terminology and the derivation of CAVI is based on Blei et al. (2017).

7.1 A finite-dimensional parameterization of PACFGs

This section starts by describing the latent probabilistic grammar and a dataset of observed terminal sequences with latent derivations in one uniform probabilistic model. Such a model is simply given by the specification of the joint probability of the latent variables and the observations. In the proposed model, there are no cyclic dependencies between random variables. The model is therefore an instance of a *generative probabilistic model* as described in Section 2.5. According to generative modeling, the observed sequential data is analyzed by reconstruction of plausible PACFGs and derivations that yield the observed terminal sequences.

The process of (re-)generating a dataset of terminal sequences according to a grammar model with fixed rule set consists of three steps. First, the parameters of the grammar are sampled and the rewrite probabilities are calculated. The probabilistic grammar thus generated is then used to sample a set of derivation trees from which, finally, the terminal sequences are calculated as the leafs of the trees. This three-step process is described by the factorization of the joint probability of the grammar parameters $\boldsymbol{\theta} \in \Theta$, the derivations $\bar{\boldsymbol{r}} = (\boldsymbol{r}^1, \dots, \boldsymbol{r}^I) \in \mathscr{R}^I$, and the terminal sequences $\bar{\boldsymbol{w}} = (\boldsymbol{w}^1, \dots, \boldsymbol{w}^I) \in (T^*)^I$ into three terms,

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) = p(\boldsymbol{\theta})\ p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})\ p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}), \tag{7.1}$$

where $I \in \mathbb{N}$ is the size of the dataset, $\mathscr{R} \subseteq R^*$ is the set of all derivations,

$$\mathscr{R} = \{\boldsymbol{r} \in R^* \mid \exists \boldsymbol{w} \in T^* : \boldsymbol{r}(\text{Start}) = \boldsymbol{w}\}, \tag{7.2}$$

and the parameter space Θ is left unspecified for now. The factorization can be rewritten to compute the posterior distribution of the parameters which models the plausibilities of the parameter settings after the observation of the data,

$$p(\boldsymbol{\theta} \mid \bar{\boldsymbol{w}}) = \sum_{\bar{\boldsymbol{r}} \in (R^*)^I} p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}}) \propto \sum_{\bar{\boldsymbol{r}} \in (R^*)^I} p(\boldsymbol{\theta})\ p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})\ p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}). \tag{7.3}$$

Since the exact calculation of the normalizing constant of the posterior is intractable (for all parameterizations considered in this study), variational Bayesian inference is used to approximate it as described later in Section 7.4. The following paragraphs first describe the distributions $p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}})$ and $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ in greater detail before the prior distribution over the parameters $p(\boldsymbol{\theta})$ is explained in the next section.

The factorization (e.g., the generation process) encodes the statistical dependencies and independencies of the observed data $\bar{\boldsymbol{w}}$ and the latent variables $\boldsymbol{\theta}$ and $\bar{\boldsymbol{r}}$. For example, the sequences $\bar{\boldsymbol{w}}$ are independent from the grammar parameters $\boldsymbol{\theta}$ when the derivations $\bar{\boldsymbol{r}}$ are known. In fact, the distribution $p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}})$ puts all of its probability mass on a single point which is the leaf sequence of the derivation tree,

$$p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}) = \prod_{i=1}^{I} \mathbb{1}\left(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i\right). \tag{7.4}$$

The facts that the derivations are sampled independently from the grammar and that each terminal sequence only depends on its respective derivation is not yet encoded in Equation 7.1. They can be incorporated by explicitly writing-out the probabilities $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ and $p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}})$ as products,

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) = p(\boldsymbol{\theta}) \prod_{i=1}^{I} p(\boldsymbol{r}^i \mid \boldsymbol{\theta})\; p(\boldsymbol{w}^i \mid \boldsymbol{r}^i) \tag{7.5}$$

$$= p(\boldsymbol{\theta}) \prod_{i=1}^{I} p(\boldsymbol{r}^i \mid \boldsymbol{\theta})\; \mathbb{1}\Big(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i\Big). \tag{7.6}$$

The explicit denotation of the independencies between the derivations further illustrates a distinction of the latent variables that is important for parameter inference (Blei et al., 2017): The grammar parameters $\boldsymbol{\theta}$ are *global* variables of the model, that is they are the same for all derivations. In contrast, the derivations $\bar{\boldsymbol{r}}$ are *local* variables, since there is a one-to-one relationship between derivations and terminal sequences.

A more detailed description of the distributions $p(\boldsymbol{\theta})$ and $p(\boldsymbol{r}^i \mid \boldsymbol{\theta})$ requires a parameterization of the rewrite probabilities $p_A(r)$ for $A \in N$ and $r \in R$. The most direct parameterization would be $p_A(r) = \boldsymbol{\theta}_r^A$ for

$$\boldsymbol{\theta} \in \Theta = \left\{ \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^{|N| \times |R|} \,\middle|\, \forall A \in N\colon \sum_{r \in R} \boldsymbol{\theta}_r^A = 1 \text{ and } \forall r \in R\colon \boldsymbol{\theta}_r^A > 0 \iff A \in \operatorname{dom}(r) \right\}. \tag{7.7}$$

The term $\sum_{r \in R} \boldsymbol{\theta}_r^A = 1$ ensures the normalization of $p_A(r)$ for all $A \in N$ and $\boldsymbol{\theta}_r^A > 0$ if and only if $A \in \operatorname{dom}(r)$ for all $A \in N$ and $r \in R$ is required to satisfy the axiom of a PACFG from Equation 5.22. Because of the potentially infinite cardinality of N, there is, however, no canonical choice of a distribution over Θ. In fact, any nontrivial distribution needs to exploit structural properties of the set of nonterminals N.

To overcome the problem of the direct parameterization, this study proposes to assume a surjective feature projection $\phi\colon N \to \Phi$ that projects nonterminals into a finite set of features Φ. According to such a feature projection, nonterminals that are projected to the same feature are assumed to share a distribution over rewrite functions. That is

$$\phi(A) = \phi(B) \implies p_A(r) = p_B(r) \tag{7.8}$$

for all nonterminals $A, B \in N$ and rules $r \in R$. Consequently, A and B then also share their sets of applicable rewrite function, $R_A = R_B$. This justifies the following notation of the set of rules that are applicable to nonterminals of a given feature $f \in \Phi$:

$$R_f = \{ r \in R \mid \exists A \in \operatorname{dom}(r)\colon \phi(A) = f \} \tag{7.9}$$

Nonterminal feature projections are a useful modeling tool to specify probability distributions using coarsened representations of nonterminals. A feature projection is for example used in the rhythm grammar described in Section 5.1. There, Equation 5.7 defines a rather extreme case in which the set of nonterminals is infinite (essentially the set of all positive rational numbers less or equal to 1) and ϕ simply classifies nonterminals as being the unique start symbol or not. This feature projection thus enables that the splitting of a chord duration does not depend on the duration but only on the split ratio. The opposite extreme of a feature projection that is the identity function $\phi\colon N \to N$ ($\phi(A) = A$). This is a possible choice for all grammars with a finite number of nonterminals. In particular, all standard context-free grammars fall into this class. Another kind of feature projection is presented later for the computational experiments to define a harmony grammar whose probability model is transpositionally independent. In that harmony grammar, chords of the same chord form share an identical distribution over rewrite functions. The rewrite probabilities are therefore independent of the chord root.

Let now $G = (T, N, \text{Start}, R)$ be a PACFG with a nonterminal feature projection $\phi\colon N \to \Phi$. In this case, the rewrite probabilities can be parameterized by $p_A(r) = \boldsymbol{\theta}_r^{\phi(A)}$ for

$$\boldsymbol{\theta} \in \Theta = \left\{ \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^{|\Phi| \times |R|} \;\middle|\; \forall f \in \Phi\colon \sum_{r \in R} \boldsymbol{\theta}_r^f = 1 \text{ and } \forall r \in R\colon \boldsymbol{\theta}_r^f > 0 \iff r \in R_f \right\}. \tag{7.10}$$

This definition of Θ is analogues to its counterpart for the direct parameterization shown in Equation 7.7. In contrast to the direct parameterization, this choice of Θ has the mathematical advantage of being finite, which is used in the next sections to define a distribution over Θ that allows for efficient parameter inference.

Using the parameterization $p_A(r) = \boldsymbol{\theta}_r^{\phi(A)}$, the probability of a derivation is given by

$$p(\boldsymbol{r}^i \mid \boldsymbol{\theta}) = \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\theta}_{\boldsymbol{r}_k^i}^{\phi(A_k^i)} \tag{7.11}$$

where A_k^i denotes the leftmost nonterminal in the k-th step of the derivation $\boldsymbol{r}^i$,

$$A_k^i = \textsc{leftmost}(\boldsymbol{r}_{1:k-1}^i(\text{Start})) \tag{7.12}$$

as defined in Equation 5.24. In other words, A_k^i is the nonterminal to which the rule $\boldsymbol{r}_k^i$ is applied.

Every part of the probabilistic model but the prior distribution $p(\boldsymbol{\theta})$ is now specified,

$$\begin{aligned} p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) &= p(\boldsymbol{\theta})\ p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})\ p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}) &\quad (7.13)\\ &= p(\boldsymbol{\theta}) \prod_{i=1}^{I} \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\theta}_{\boldsymbol{r}_k^i}^{\phi(A_k^i)}\ \mathbb{1}\Big(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i\Big). &\quad (7.14) \end{aligned}$$

One more generic assumption is put on the distribution of the parameters $p(\boldsymbol{\theta})$: The rewrite probabilities of the grammar are generated independently for each feature. That is

$$p(\boldsymbol{\theta}) = \prod_{f \in \Phi} p(\boldsymbol{\theta}^f). \tag{7.15}$$

Therefore,

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) = \left(\prod_{f \in \Phi} p(\boldsymbol{\theta}^f) \right) \prod_{i=1}^{I} \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\theta}^{\phi(A^i_k)}_{\boldsymbol{r}^i_k} \, \mathbb{1}(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i). \tag{7.16}$$

A visualization of the model as a Bayesian network is shown in Figure 7.1. The next two sections present and discuss the choice of $p(\boldsymbol{\theta}^f)$.

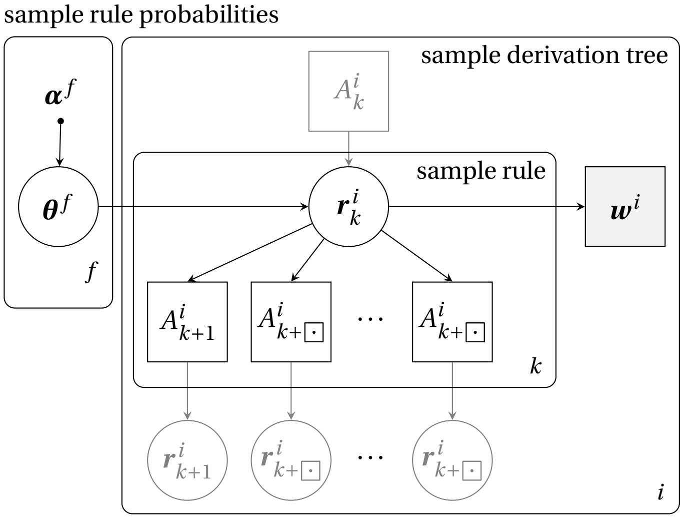


Figure 7.1 – Bayesian generative model of rule probabilities $\boldsymbol{\theta}$, derivations $\boldsymbol{r}^i$, and sequences $\boldsymbol{w}^i$ ($i \in \{1, \dots, I\}$). Circles denote random choices and squares denote deterministic calculations from random choices. The arrows show the dependencies between the random variables. The left plate shows that for each feature $f \in \Phi$, the probabilities $\boldsymbol{\theta}^f$ of a rule being applied to a nonterminal $A \in N$ with feature $\phi(A) = f$ are drawn independently. The inner right plate shows a recursive cell of the tree sampling process in which the rule $\boldsymbol{r}^i_k$ is applied to the nonterminal A^i_k (A^i_1 = Start, $\boldsymbol{r}^i_k$ denotes the k-th rule of derivation $\boldsymbol{r}^i$, $k \in \{1, \dots, |\boldsymbol{r}^i|\}$). The grey nodes show how the recursive cells are connected. The placeholder $\boxdot$ is used to denote indices that depend on random choices of their left siblings. The outer right plate shows that the observed sequence $\boldsymbol{w}^i$ is deterministically calculated from $\boldsymbol{r}^i$.

7.2 The choice of the prior distribution

The rational belief about the grammar parameters prior to the observation is represented by the distribution $p(\boldsymbol{\theta})$. Since the concrete value of $\boldsymbol{\theta}$ is unknown, the rational belief about $\boldsymbol{\theta}$ changes with the observation of derivations $\bar{\boldsymbol{r}} = (\boldsymbol{r}^1, \dots, \boldsymbol{r}^I)$. The full model does observe terminal sequences, not their derivations, but in this section it is assumed that the derivations themselves are observed to discuss the construction of the model. Bayes' rule formalizes the observation of derivations by

$$p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}}) \propto p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta}), \tag{7.17}$$

where the distribution $p(\boldsymbol{\theta})$ represents the *prior* belief about $\boldsymbol{\theta}$, the term $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ represents the model for (re-)generating the observed data, and the distribution $p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}})$ represents the belief about $\boldsymbol{\theta}$ as changed after the observation (the *posterior* belief). The observation process is thus modeled as a multiplication of the prior belief about $\boldsymbol{\theta}$ with the probability that the observed data is generated using $\boldsymbol{\theta}$. The function $\boldsymbol{\theta} \mapsto p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ is also called the *likelihood function* and the value $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ is called the *likelihood* of $\boldsymbol{\theta}$ given the data $\bar{\boldsymbol{r}}$. The terms likelihood function and data-generating distribution or sample distribution are often used interchangeably.

Before the full observation process is described, we focus on how the observation of a single rule $r \in R_f$ being applied to a nonterminal with feature f changes the rational belief about $\boldsymbol{\theta}$. Using the parameterization $p_A(r) = \boldsymbol{\theta}_r^{\phi(A)}$ proposed in the last section, the observation only changes the belief about the part of $\boldsymbol{\theta}$ that represents the distribution of rules that are applied to nonterminals with feature f — the categorical distribution represented by $\boldsymbol{\theta}^f$. The observation of the rule r is therefore fully described by

$$p(\boldsymbol{\theta}^f \mid r) \propto p(r \mid \boldsymbol{\theta}^f)\ p(\boldsymbol{\theta}^f) = \boldsymbol{\theta}_r^f\ p(\boldsymbol{\theta}^f). \tag{7.18}$$

The random variable $\boldsymbol{\theta}^f \in \mathbb{R}_{>0}^{|R_r|}$ is a $|R_f|$-dimensional vector of positive real numbers that sum to one. Such vectors and categorical distributions $p(r \mid \boldsymbol{\theta}^f)$ stand in a one-to-one relation. The choice of a distribution over $\boldsymbol{\theta}^f$ is therefore a choice of a distribution over distributions. In Bayesian statistics, the default choice for a distribution over categorical distributions $\boldsymbol{\theta}^f$ is the *Dirichlet distribution.* Its density function is:

$$p(\boldsymbol{\theta}^f) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha}^f)} \prod_{r \in R_f} (\boldsymbol{\theta}_r^f)^{\boldsymbol{\alpha}_r^f - 1} \tag{7.19}$$

The parameter vector $\boldsymbol{\alpha}^f \in \mathbb{R}_{>0}^{|R_f|}$ stores a positive real number $\boldsymbol{\alpha}_r^f$ for each rule $r \in R_f$.[1] The term $\mathrm{B}(\boldsymbol{\alpha}^f)$ is the normalizing constant of the distribution represented by the multivariate

[1] Note that the notation of such parameter vectors clashes with the notation of sequences consisting of terminals and nonterminals. This is not expected to be a problem, because of the rather diverse nature of those objects.

beta function B:

$$\mathrm{B}(\boldsymbol{\alpha}^f) = \int_{\boldsymbol{\alpha}^f \in \mathbb{R}_{>0}^{|R_f|}} \prod_{r \in R_f} (\boldsymbol{\theta}_r^f)^{\boldsymbol{\alpha}_r^f - 1} \tag{7.20}$$

The modeling power of the Dirichlet distribution results from the fact that the observation of data is representable by a simple change of the distribution's parameter vector. More concretely, if $\boldsymbol{\alpha}^f$ denotes the parameter vector of the prior distribution $p(\boldsymbol{\theta}^f)$, then the parameter vector of the posterior distribution $p(\boldsymbol{\theta}^f \mid r)$ is $\hat{\boldsymbol{\alpha}}^f$ where

$$\hat{\boldsymbol{\alpha}}_{r'}^f = \boldsymbol{\alpha}_{r'}^f + \mathbb{1}(r = r') \tag{7.21}$$

for all $r' \in R_f$, because:

$$\begin{align}
p(\boldsymbol{\theta}^f \mid r) &\propto \boldsymbol{\theta}_r^f \; p(\boldsymbol{\theta}^f) \tag{7.22}\\
&\propto \boldsymbol{\theta}_r^f \prod_{r' \in R_f} (\boldsymbol{\theta}_{r'}^f)^{\boldsymbol{\alpha}_{r'}^f - 1} \tag{7.23}\\
&= \prod_{r' \in R_r} (\boldsymbol{\theta}_{r'}^f)^{\mathbb{1}(r=r')} (\boldsymbol{\theta}_{r'}^f)^{\boldsymbol{\alpha}_{r'}^f - 1} \tag{7.24}\\
&= \prod_{r' \in R_r} (\boldsymbol{\theta}_{r'}^f)^{\boldsymbol{\alpha}_{r'}^f + \mathbb{1}(r=r') - 1} \tag{7.25}
\end{align}$$

Equation 7.21 suggests an intuitive interpretation of the parameter vector $\boldsymbol{\alpha}^f$ of the Dirichlet distribution $p(\boldsymbol{\theta}^f)$. This vector essentially stores the number of how often each rule was observed. It is thus also called a *pseudocount vector*. With this intuition, the definition of the Dirichlet distribution in Equation 7.19 implies in particular that higher rewrite probabilities $\boldsymbol{\theta}_r^f$ become more plausible as more applications of the rule r to nonterminals with feature f are observed.

Using the notation $\boldsymbol{\theta}^f \sim \mathrm{Dir}(\boldsymbol{\alpha}^f)$ for the fact that $\boldsymbol{\theta}^f$ is Dirichlet distributed with pseudocount vector $\boldsymbol{\alpha}^f$, the observation of the rule r leads to the implication

$$\boldsymbol{\theta}^f \sim \mathrm{Dir}(\boldsymbol{\alpha}^f) \implies \boldsymbol{\theta}^f \mid r \sim \mathrm{Dir}(\hat{\boldsymbol{\alpha}}^f). \tag{7.26}$$

Such prior distributions $p(\boldsymbol{\theta}^f)$ for which the posterior $p(\boldsymbol{\theta}^f \mid r)$ is in the same distribution family are called *conjugate priors* for their data-generating distributions $p(r \mid \boldsymbol{\theta}^f)$ (Bishop, 2006; Berger, 2013). The presented argumentation thus proves that Dirichlet distributions are conjugate priors for categorical distributions.

Now by assuming $\boldsymbol{\theta}^f$ to be Dirichlet distributed with parameter vector $\boldsymbol{\alpha}^f$ for all features $f \in \Phi$, the prior distribution $p(\boldsymbol{\theta})$ has the form

$$p(\boldsymbol{\theta}) = \prod_{f \in \Phi} \frac{1}{\mathrm{B}(\boldsymbol{\alpha}^f)} \prod_{r \in R_f} (\boldsymbol{\theta}^f_r)^{\boldsymbol{\alpha}^f_r - 1}. \tag{7.27}$$

Extending Equation 7.16, the full generative model of the parameters $\boldsymbol{\theta}$, derivations $\bar{\boldsymbol{r}}$, and terminal sequences $\bar{\boldsymbol{w}}$ is thus completely specified by the factorization

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) = \left(\prod_{f \in \Phi} \frac{1}{\mathrm{B}(\boldsymbol{\alpha}^f)} \prod_{r \in R_f} (\boldsymbol{\theta}^f_r)^{\boldsymbol{\alpha}^f_r - 1} \right) \prod_{i=1}^{I} \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\theta}^{\phi(A^i_k)}_{\boldsymbol{r}^i_k} \, \mathbb{1}\left(\boldsymbol{r}^i(\mathrm{Start}) = \boldsymbol{w}^i \right). \tag{7.28}$$

The next section shows that the choice of the prior distribution in Equation 7.27 makes $p(\boldsymbol{\theta})$ a conjugate prior for $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$.

7.3 Exponential families

Conjugate priors are in general neither unique nor do they exist for all data-generating distributions (likelihood functions). The key point in the proof from the last section — that Dirichlet distributions are conjugate priors for categorical distributions — is the rewriting trick in Equation 7.24:

$$\boldsymbol{\theta}^f_r = \prod_{r' \in R_r} (\boldsymbol{\theta}^f_{r'})^{\mathbb{1}(r = r')} \tag{7.29}$$

This identity reduces the multiplication of the likelihood function with the prior probability to a summation in the exponent of the parameter. The trick is generalizable to a class of distributions called exponential families. In particular, likelihood functions in this class are proven to always have a conjugate prior.

Exponential family Let x be a random variable with sample space X, let Λ be the parameter space of its distribution family and let $\lambda \in \Lambda$ be the parameter of the distribution of x. The distribution family of x is called an *exponential family* if its sample space does not change with different parameters and if its density function[2] has the form

$$p(x) = h(x)\ \exp\left[\langle \eta(\lambda), t(x) \rangle - a(\lambda) \right] \tag{7.30}$$

[2] The term *density function* is meant here as a general concept to refer to both probability density functions of continuous random variables and probability mass functions of discrete random variables.

for functions

$$
\begin{aligned}
h &\colon X \to \mathbb{R}_{>0} && \text{(base measure)} && (7.31)\\
a &\colon \Lambda \to \mathbb{R} && \text{(log-partition function)} && (7.32)\\
\eta &\colon \Lambda \to \mathbb{R}^d && \text{(natural parameter)} && (7.33)\\
t &\colon X \to \mathbb{R}^d && \text{(sufficient statistics)} && (7.34)
\end{aligned}
$$

where $\langle \eta(\lambda), t(x) \rangle = \sum_{j=1}^{d} \eta(\lambda)_j \, t(x)_j$ denotes the Euclidean scalar product of $\mathbb{R}^d$ (also called dot product or inner product).

One core idea of exponential families is the separability of the density function into terms that only depend either on the value of x or the parameter λ. Maybe surprisingly at first, this separation is possible for most of the common distribution families such as binomial, categorical, multinomial, Gaussian, and Dirichlet distributions. The derivation-generating distribution $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ is also of this from for

$$
\begin{aligned}
h(\bar{\boldsymbol{r}}) &= 1 && (7.35)\\
a(\boldsymbol{\theta}) &= 0 && (7.36)\\
\eta(\boldsymbol{\theta})_r^f &= \log \boldsymbol{\theta}_r^f && (7.37)\\
t(\bar{\boldsymbol{r}})_r^f &= \sum_{i=1}^{I} \sum_{k=1}^{|\boldsymbol{r}^i|} \mathbb{1}\Big(f = \phi(A_k^i)\Big) \mathbb{1}\Big(r = \boldsymbol{r}_k^i\Big) && (7.38)
\end{aligned}
$$

where $\bar{\boldsymbol{r}} \in \mathscr{R}^I$ (defined in Equation 7.2), $\boldsymbol{\theta} \in \Theta$ (defined in Equation 7.10), $f \in \Phi$, and $r \in R$, because the sample space $\mathscr{R}^I$ does not depend on $\boldsymbol{\theta}$ and

$$
\begin{aligned}
p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta}) &= \exp\left[\log \prod_{i=1}^{I} \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\theta}_{\boldsymbol{r}_k^i}^{\phi(A_k^i)}\right] && (7.39)\\
&= \exp\left[\sum_{\substack{f\in\Phi\\ r\in R}} \log \boldsymbol{\theta}_r^f \sum_{i=1}^{I} \sum_{k=1}^{|\boldsymbol{r}^i|} \mathbb{1}\Big(f = \phi(A_k^i)\Big) \mathbb{1}\Big(r = \boldsymbol{r}_k^i\Big)\right] && (7.40)\\
&= \exp \langle \quad \eta(\boldsymbol{\theta}) \ , t(\bar{\boldsymbol{r}}) \qquad\qquad\qquad \rangle. && (7.41)
\end{aligned}
$$

Note the similarity of Equation 7.40 and the rewrite trick for single categorical distributions in Equation 7.24.

The sufficient statistics $t(x)$ of an exponential family have a very intuitive interpretation. They project a value x to exactly the information that is needed to calculate the probability $p(x)$. For instance, the sufficient statistics of $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ give the total number of how many times a rule $r \in R$ is applied to a nonterminal with a feature $f \in \Phi$ in the derivations $\boldsymbol{r}^1, \ldots, \boldsymbol{r}^I \in \mathscr{R}$. Since the probability of a derivation does not depend on the order of the rules, the sufficient statistics abstract from this order by returning only the number of each rule application type. This abstraction is, however, only possible because the sample space is restricted to sequences of rules that are derivations of some sequence.

The roles of the other three functions from the definition of exponential families are rather technical. The base measure h can constantly weigh some values x higher than others, the natural parameter is the parameter transformation that enables the separation from the sufficient statistics, and the log-partition function is simply the logarithm of the normalizing constant of the distribution,

$$a(\lambda) = \log \int_{x \in X} h(x) \exp \langle \eta(\lambda), t(x) \rangle . \tag{7.42}$$

The prior distribution $p(\boldsymbol{\theta})$ belongs also to an exponential family with

$$\begin{align} h(\boldsymbol{\theta}) &= 1 \tag{7.43} \\ a(\boldsymbol{\alpha}) &= \sum_{f \in \Phi} \log \mathrm{B}(\boldsymbol{\alpha}^f) \tag{7.44} \\ \eta(\boldsymbol{\alpha})_r^f &= \boldsymbol{\alpha}_r^f - 1 \tag{7.45} \\ t(\boldsymbol{\theta})_r^f &= \log \boldsymbol{\theta}_r^f \tag{7.46} \end{align}$$

where $\boldsymbol{\theta} \in \Theta$ (defined in Equation 7.10), $\boldsymbol{\alpha} \in \mathbb{R}_{>0}^{|\Phi| \times |R|}$, $f \in \Phi$, and $r \in R$, because the sample space Θ does not depend on $\boldsymbol{\alpha}$ and

$$\begin{align} p(\boldsymbol{\theta}) &= \exp \left[\log \prod_{f \in \Phi} \frac{1}{\mathrm{B}(\boldsymbol{\alpha}^f)} \prod_{r \in R_f} (\boldsymbol{\theta}_r^f)^{\boldsymbol{\alpha}_r^f - 1} \right] \tag{7.47} \\ &= \exp \left[\sum_{f \in \Phi} \sum_{r \in R} (\boldsymbol{\alpha}_r^f - 1) \log \boldsymbol{\theta}_r^f - \sum_{f \in \Phi} \log \mathrm{B}(\boldsymbol{\alpha}^f) \right] \tag{7.48} \\ &= \exp \left[\langle \quad \eta(\boldsymbol{\alpha}) \quad , t(\boldsymbol{\theta}) \ \rangle - a(\boldsymbol{\alpha}) \qquad \right] . \tag{7.49} \end{align}$$

A comparison of the exponential forms of the likelihood $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$ reveals that the natural parameter of the likelihood is equal to the sufficient statistics of the prior, $\eta(\boldsymbol{\theta}) = t(\boldsymbol{\theta})$. This prior is therefore a conjugate prior, because

$$\begin{align} p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}}) &\propto p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta}) \, p(\boldsymbol{\theta}) \tag{7.50} \\ &\propto \exp \left[\langle \eta(\boldsymbol{\theta}), t(\bar{\boldsymbol{r}}) \rangle + \langle \eta(\boldsymbol{\alpha}), t(\boldsymbol{\theta}) \rangle \right] \tag{7.51} \\ &= \exp \langle \eta(\boldsymbol{\alpha}) + t(\bar{\boldsymbol{r}}), t(\boldsymbol{\theta}) \rangle \tag{7.52} \\ &= \exp \langle \eta(\boldsymbol{\alpha} + t(\bar{\boldsymbol{r}})), t(\boldsymbol{\theta}) \rangle \tag{7.53} \end{align}$$

by symmetry and linearity in the first component of the scalar product. Since the distribution family of $p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}})$ is known, its density can be computed exactly in exponential and conventional form,

$$p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}}) = \exp\left[\langle \eta(\hat{\boldsymbol{\alpha}}), t(\boldsymbol{\theta})\rangle - a(\hat{\boldsymbol{\alpha}})\right] = \prod_{f\in\Phi} \frac{1}{\mathrm{B}(\hat{\boldsymbol{\alpha}}^f)} \prod_{r\in R_f} (\hat{\boldsymbol{\theta}}_r^f)^{\hat{\boldsymbol{\alpha}}_r^f - 1}, \tag{7.54}$$

where $\hat{\boldsymbol{\alpha}}_r^f = \boldsymbol{\alpha}_r^f + \sum_{i=1}^{I} \sum_{k=1}^{|\boldsymbol{r}^i|} \mathbb{1}\big(f = \phi(A_k^i)\big)\, \mathbb{1}\big(r = \boldsymbol{r}_k^i\big)$.

7.4 Coordinate ascent variational inference (CAVI)

The full model is specified as a factorization of the joint distribution of parameters $\boldsymbol{\theta}$, derivations $\bar{\boldsymbol{r}}$, and terminal sequences $\bar{\boldsymbol{w}}$ in Equation 7.28. The conditional distribution $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ of parameters $\boldsymbol{\theta}$ and derivations $\bar{\boldsymbol{r}}$ given the dataset of observed sequences $\bar{\boldsymbol{w}}$ inherits all information that can be learned from this observation under assumptions of the model. For instance, the marginal distribution $p(\boldsymbol{\theta} \mid \bar{\boldsymbol{w}}) = \sum_{\bar{\boldsymbol{r}}\in\mathscr{R}^I} p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ can be used to compute the probability of a new unseen sequence $\boldsymbol{v} \in T^*$,

$$p(\boldsymbol{v} \mid \bar{\boldsymbol{w}}) = \int_{\boldsymbol{\theta}\in\Theta} p(\boldsymbol{v}, \boldsymbol{\theta} \mid \bar{\boldsymbol{w}}) = \int_{\boldsymbol{\theta}\in\Theta} p(\boldsymbol{v} \mid \boldsymbol{\theta}, \bar{\boldsymbol{w}})\; p(\boldsymbol{\theta} \mid \bar{\boldsymbol{w}}). \tag{7.55}$$

The calculation of the normalizing constant $p(\bar{\boldsymbol{w}})$ of the posterior distribution

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}}) = \frac{p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}})}{p(\bar{\boldsymbol{w}})} \tag{7.56}$$

is, however, intractable. This section thus presents an approximation method to compute $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ based on variational Bayesian inference.

The idea of variational Bayesian inference is to approximate the distribution $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ with a distribution $q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})$ from a simpler family. The family of the considered approximations is thereby chosen to closely resemble the original distribution specified by the generative model. This study approximates $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ by assuming independence between the parameters $\boldsymbol{\theta}$ and the derivations $\bar{\boldsymbol{r}}$,

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}}) \approx q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) = q(\boldsymbol{\theta})\; q(\bar{\boldsymbol{r}}), \tag{7.57}$$

where the distributions $q(\boldsymbol{\theta})$ and $q(\bar{\boldsymbol{r}})$ are chosen to be in the same class as $p(\boldsymbol{\theta})$ and $p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})$, respectively. That is, $q(\boldsymbol{\theta})$ is a product of Dirichlet distributions with pseudocount vectors $\tilde{\boldsymbol{\alpha}}^f$ over features $f \in \Phi$,

$$q(\boldsymbol{\theta}) = \prod_{f\in\Phi} \frac{1}{\mathrm{B}(\tilde{\boldsymbol{\alpha}}^f)} \prod_{r\in R_f} (\boldsymbol{\theta}_r^f)^{\tilde{\alpha}_r^f}, \tag{7.58}$$

and $q(\bar{\boldsymbol{r}})$ is a distribution over derivations with parameters $\tilde{\boldsymbol{\theta}}$,

$$q(\bar{\boldsymbol{r}}) = \prod_{i=1}^{I} \prod_{k=1}^{|\boldsymbol{r}^i|} \tilde{\boldsymbol{\theta}}_{\boldsymbol{r}_k^i}^{\phi(A_k^i)}. \tag{7.59}$$

This kind of approximation that only assumes independence of a model's latent variables by keeping the distribution families for each variable, is called a *mean-field assumption* in the literature (Blei et al., 2017).

The goal of inference is to find a distribution $q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})$ in the family of considered approximations that is as similar to the approximated distribution $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$. The similarity between the distributions $q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})$ and $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ is measured by their *Kullback-Leibler divergence* (KL divergence),

$$\mathrm{KL}\big(q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) \,\|\, p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})\big) = \mathbb{E}_{q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})}\left[\log q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) - \log p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})\right], \tag{7.60}$$

which is the expected logarithmic difference. This KL divergence cannot be computed directly, because it requires the computation of the probability $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$. Fortunately, the unknown normalizing constant $p(\bar{\boldsymbol{w}})$ of the distribution $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$, also called the *evidence* for the model, can be moved out of the expectation,

$$\mathbb{E}_{q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})}\left[\log q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) - \log p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})\right] = \mathbb{E}_{q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})}\left[\log q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) - \log p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}})\right] + \log p(\bar{\boldsymbol{w}}), \tag{7.61}$$

which implies

$$\mathrm{KL}\big(q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) \,\|\, p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})\big) + \mathbb{E}_{q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) - \log q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})\right] = \log p(\bar{\boldsymbol{w}}). \tag{7.62}$$

Since the KL divergence is always positive or equal to zero (Kullback and Leibler, 1951), the term

$$\mathbb{E}_{q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) - \log q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})\right] \tag{7.63}$$

is called the *Evidence Lower BOund* (ELBO). Minimizing the KL divergence is therefore equivalent to maximizing the ELBO.

The ELBO can be iteratively improved by maximization with respect to one latent variable at a time. This procedure is called *Coordinate Ascent Variational Inference* (CAVI; see e.g., Bishop, 2006). The optimal coordinate ascent update with respect to $\boldsymbol{\theta}$ is derived analytically by rewriting the ELBO with constant $\bar{\boldsymbol{r}}$:

$$\begin{aligned}
&\mathbb{E}_{q(\boldsymbol{\theta},\bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta},\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})-\log q(\boldsymbol{\theta},\bar{\boldsymbol{r}})\right]\\
&=\mathbb{E}_{q(\boldsymbol{\theta})}\left[\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta},\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})\right]-\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\log q(\boldsymbol{\theta})+\log q(\bar{\boldsymbol{r}})\right]\right] && (7.64)\\
&=\mathbb{E}_{q(\boldsymbol{\theta})}\left[\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta},\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})\right]-\log q(\boldsymbol{\theta})\right]-\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\log q(\bar{\boldsymbol{r}})\right] && (7.65)\\
&=\mathbb{E}_{q(\boldsymbol{\theta})}\left[\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta}\mid\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})\right]-\log q(\boldsymbol{\theta})\right]+\underbrace{\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\log p(\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})-\log q(\bar{\boldsymbol{r}})\right]}_{\text{constant}} && (7.66)
\end{aligned}$$

Here, Equation 7.64 iterates the expectation and applies the mean-field assumption (Equation 7.59). Equations 7.65 and 7.66 use the linearity of the expected value and the definition of conditional probability. The next step is the creative part of the mathematical derivation. It is easy to understand that it works, but it might look like magic first. By setting

$$q^*(\boldsymbol{\theta})\propto\exp\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta}\mid\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})\right],\tag{7.67}$$

it follows that the ELBO is equal to the negative KL divergence between $q(\boldsymbol{\theta})$ and $q^*(\boldsymbol{\theta})$ plus an additive constant,

$$\mathbb{E}_{q(\boldsymbol{\theta},\bar{\boldsymbol{r}})}\left[\log p(\boldsymbol{\theta},\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})-\log q(\boldsymbol{\theta},\bar{\boldsymbol{r}})\right]=-\mathrm{KL}\left(q(\boldsymbol{\theta})\parallel q^*(\boldsymbol{\theta})\right)+\text{constant term.}\tag{7.68}$$

Maximizing the ELBO with respect to $\boldsymbol{\theta}$ is thus equivalent to minimizing $\mathrm{KL}\left(q(\boldsymbol{\theta})\parallel q^*(\boldsymbol{\theta})\right)$. Since the KL divergence cannot be negative, it is minimized if its arguments are equal. The optimal coordinate ascent update with respect to $\boldsymbol{\theta}$ by fixed $\bar{\boldsymbol{r}}$ is therefore achieved by setting $q(\boldsymbol{\theta}):=q^*(\boldsymbol{\theta})$. The analogue update with respect to $\bar{\boldsymbol{r}}$ is given by

$$q^*(\bar{\boldsymbol{r}})\propto\exp\mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p(\bar{\boldsymbol{r}}\mid\boldsymbol{\theta},\bar{\boldsymbol{w}})\right].\tag{7.69}$$

Equations 7.67 and 7.69 state the optimal updates of the distributions. The following calculation derives the update equations for the variational parameters (e.g., the parameters of the approximating distributions $q(\boldsymbol{\theta})$ and $q(\bar{\boldsymbol{r}})$). Since

$$p(\boldsymbol{\theta}\mid\bar{\boldsymbol{r}},\bar{\boldsymbol{w}})\propto\exp\left\langle\eta(\boldsymbol{\alpha})+t(\bar{\boldsymbol{r}}),t(\boldsymbol{\theta})\right\rangle\tag{7.70}$$

by Equation 7.53, the updated distribution $q^*(\boldsymbol{\theta})$ is in the same exponential family as $q(\boldsymbol{\theta})$,

$$\begin{aligned}
q^*(\boldsymbol{\theta})&\propto\exp\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\left\langle\eta(\boldsymbol{\alpha})+t(\bar{\boldsymbol{r}}),t(\boldsymbol{\theta})\right\rangle\right] && (7.71)\\
&=\exp\left\langle\eta(\boldsymbol{\alpha})+\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[t(\bar{\boldsymbol{r}})\right],t(\boldsymbol{\theta})\right\rangle. && (7.72)
\end{aligned}$$

The coordinate ascent parameter update for the distribution $q(\boldsymbol{\theta})$ is therefore

$$\tilde{\boldsymbol{\alpha}}_r^f:=\boldsymbol{\alpha}_r^f+\mathbb{E}_{q(\bar{\boldsymbol{r}})}\left[\sum_{i=1}^{I}\sum_{k=1}^{|\boldsymbol{r}^i|}\mathbb{1}\left(f=\phi(A_k^i)\right)\mathbb{1}\left(r=\boldsymbol{r}_k^i\right)\right],\tag{7.73}$$

where the expected value is approximated using the Monte-Carlo estimate in Equation 6.36.

Note that $\boldsymbol{\alpha}$ denotes the parameter of the prior distribution $p(\boldsymbol{\theta})$ and $\tilde{\boldsymbol{\alpha}}$ denotes the parameter of the approximation distribution $q(\boldsymbol{\theta})$.

Analogously to Equation 7.70,

$$p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta}, \bar{\boldsymbol{w}}) \propto \left(\prod_{i=1}^{I} \mathbb{1}\left(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i \right) \right) \exp \left\langle \eta(\boldsymbol{\theta}), t(\bar{\boldsymbol{r}}) \right\rangle \tag{7.74}$$

by Equation 7.41. The updated distribution $q^*(\bar{\boldsymbol{r}})$ is thus in the same exponential family as $q(\bar{\boldsymbol{r}})$,

$$\begin{aligned} q^*(\bar{\boldsymbol{r}}) &\propto \exp \mathbb{E}_{q(\boldsymbol{\theta})} \left[\log \prod_{i=1}^{I} \mathbb{1}\left(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i \right) + \langle \eta(\boldsymbol{\theta}), t(\bar{\boldsymbol{r}}) \rangle \right] &(7.75)\\ &= \left(\prod_{i=1}^{I} \mathbb{1}\left(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i \right) \right) \exp \left\langle \mathbb{E}_{q(\boldsymbol{\theta})} \left[\eta(\boldsymbol{\theta}) \right], t(\bar{\boldsymbol{r}}) \right\rangle . &(7.76) \end{aligned}$$

The coordinate ascent parameter update for the distribution $q(\bar{\boldsymbol{r}})$ is therefore

$$\tilde{\boldsymbol{\theta}}_r^f := \exp \mathbb{E}_{q(\boldsymbol{\theta})} \left[\log \boldsymbol{\theta}_r^f \right] = \frac{\exp \gamma \left(\tilde{\boldsymbol{\alpha}}_r^f \right)}{\exp \gamma \left(\sum_{r \in R_f} \tilde{\boldsymbol{\alpha}}_r^f \right)}, \tag{7.77}$$

where

$$\gamma \colon \mathbb{R}_{>0} \to \mathbb{R}_{>0}, \qquad \gamma(x) = \frac{d}{dx} \log \Gamma(x), \tag{7.78}$$

denotes the *digamma* function.[3] Thereby, Equation 7.77 follows from the fact that $q(\boldsymbol{\theta}^f)$ is a Dirichlet distribution with pseudocount vector $\tilde{\boldsymbol{\alpha}}^f$.

The alternating CAVI updates of the variational parameters $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\theta}}$ given by Equations 7.73 and 7.77 converge to a local minimum of the KL divergence in Equation 7.60. The inference procedure should therefore be performed multiple times with different start values. CAVI is furthermore easily generalizable to stochastic variational inference that uses batch updates instead of the whole dataset to calculate the expected value in Equation 7.73 (Hoffman et al., 2013).

7.5 Summary

Given an ACFG $G = (T, N, \text{Start}, R)$ with a feature projection $\phi \colon N \to \Phi$ and a dataset of terminal sequences $\bar{\boldsymbol{w}} = (\boldsymbol{w}^1, \ldots, \boldsymbol{w}^I) \in (T^*)^I$, this study proposes to assume that nonterminals with equal features share a distribution over applicable rewrite functions. The full generative model

[3] The digamma function is usually denoted by ψ in the literature. In this study, it is denoted by γ to avoid unnecessary notation overloading.

of the grammar’s parameters

$$\Theta = \left\{ \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^{|\Phi|\times|R|} \;\middle|\; \forall f \in \Phi\colon \sum_{r\in R} \boldsymbol{\theta}_r^f = 1 \text{ and } \forall r \in R\colon \boldsymbol{\theta}_r^f > 0 \iff r \in R_f \right\}, \tag{7.79}$$

and the derivations

$$\mathscr{R} = \{ \boldsymbol{r} \in R^* \mid \exists \boldsymbol{w} \in T^* : \boldsymbol{r}(\text{Start}) = \boldsymbol{w} \} \tag{7.80}$$

of the observed sequences $\bar{\boldsymbol{w}}$ is defined by the joint distribution

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) = p(\boldsymbol{\theta})\ p(\bar{\boldsymbol{r}} \mid \boldsymbol{\theta})\ p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}) \tag{7.81}$$

$$= \left(\prod_{f\in\Phi} p(\boldsymbol{\theta}^f) \right) \prod_{i=1}^{I} \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\theta}_{\boldsymbol{r}_k^i}^{\phi(A_k^i)}\ \mathbb{1}(\boldsymbol{r}^i(\text{Start}) = \boldsymbol{w}^i), \tag{7.82}$$

where $p_A(r) = \boldsymbol{\theta}_r^{\phi(A)}$.

The distribution $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ represents the information that the model can learn from the observed sequences $\bar{\boldsymbol{w}}$. Since the computation of the normalizing constant of this distribution is intractable to compute, the distribution is approximated by a distribution from a simpler family which assumes independence of $\boldsymbol{\theta}$ and $\bar{\boldsymbol{r}}$,

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}}) \approx q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) = q(\boldsymbol{\theta})\ q(\bar{\boldsymbol{r}}), \tag{7.83}$$

where:

$$q(\boldsymbol{\theta}) = \prod_{f\in\Phi} \frac{1}{\mathrm{B}(\tilde{\boldsymbol{\alpha}}^f)} \prod_{r\in R_f} (\boldsymbol{\theta}_r^f)^{\tilde{\boldsymbol{\alpha}}_r^f} \tag{7.84}$$

$$q(\bar{\boldsymbol{r}}) = \prod_{i=1}^{I} \prod_{k=1}^{|\boldsymbol{r}^i|} \tilde{\boldsymbol{\theta}}_{\boldsymbol{r}_k^i}^{\phi(A_k^i)}. \tag{7.85}$$

The optimal parameters $\tilde{\boldsymbol{\alpha}}$ and $\tilde{\boldsymbol{\theta}}$ of the approximation $q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})$ are computed by alternating coordinate ascent updates

$$\tilde{\boldsymbol{\alpha}}_r^f := \boldsymbol{\alpha}_r^f + \mathbb{E}_{q(\bar{\boldsymbol{r}})} \left[\sum_{i=1}^{I} \sum_{k=1}^{|\boldsymbol{r}^i|} \mathbb{1}\left(f = \phi(A_k^i) \right) \mathbb{1}\left(r = \boldsymbol{r}_k^i \right) \right] \tag{7.86}$$

$$\tilde{\boldsymbol{\theta}}_r^f := \frac{\exp \gamma \left(\tilde{\boldsymbol{\alpha}}_r^f \right)}{\exp \gamma \left(\sum_{r\in R_f} \tilde{\boldsymbol{\alpha}}_r^f \right)} \tag{7.87}$$

where the expected value is approximated using the Monte-Carlo estimate in Equation 6.36 and γ denotes the digamma function. These iterative updates lead to a locally optimal approximation. The inference procedure is therefore performed multiple times with varying start values.

Computational Experiments Part III

8 Grammar models for harmony and rhythm

The computational experiments described in the following two chapters use three grammar models for harmony and one for rhythm. Those models are applied both as product grammar models and as single-component models of either harmony or rhythm. This chapter describes the four component grammars in detail. All of them are in a (weakly) generalized Chomsky-normal form where each rule $r \in R$ is classified into one of three types:

Start rules which rewrite the start symbol into a nonterminal, $r : N \twoheadrightarrow N$ with $\mathrm{dom}(r) = \{\mathrm{Start}\}$,

Terminal rules which rewrite nonterminals into terminals, $r : N \twoheadrightarrow T$, and

Binary rules which rewrite nonterminals into sequences of two nonterminals, $r : N \twoheadrightarrow N^2$.

The binary rules can be further broken down into three types:

Duplication rules for which $r(A) = A\ A$ for all $A \in \mathrm{dom}(r)$,

Left-headed rules for which $r(A) = A\ B$ for all $A \in \mathrm{dom}(r)$ and some $B \in N$ and,

Right-headed rules for which $r(A) = B\ A$ for all $A \in \mathrm{dom}(r)$ and some $B \in N$,

where the nonterminal B depends on A. Rules that are either left- or right-headed are simply referred to as *headed*. As described in Chapters 1 and 4, duplication rules and headed rules are of particular importance for music modeling, because they formalize the music-theoretic concepts of prolongation and elaboration.[1]

All of the proposed grammars essentially allow for all possible headed rules and duplication rules. The inference of the rewrite probabilities from data then leads to grammars in which

[1] Duplication rules correspond to strong prolongations. A weak prolongation is either a left- or a right-headed rule.

most of the probability mass is shared by only a few rules. In other words, instead of strictly excluding certain grammar rules, *smoothed* grammars are used in which those rules that would have been excluded in expert-created rule sets have a very low but positive probability in the grammars inferred from data. Smoothed grammars are preferred over expert-created grammars, because the main interest of this study is grammar learning. By using large sets of rules which are possible prior to any observation, grammar learning can be simulated as inference of rewrite probabilities as described in the previous chapter. Chapter 9 considers supervised learning, for which the observations are the expert-created trees of the Jazz harmony treebank (see Chapter 4). Chapter 10 considers unsupervised learning, in which only chord sequences are observed and the derivation trees are inferred together with the rewrite probabilities.

For each grammar, a feature projection $\phi\colon N \to \Phi$ is used to define the parameterization of its probability model. As introduced in Section 7.1 with Equation 7.8, a feature projection is a surjective mapping from the set of nonterminals N to a set of feature values Φ such that all nonterminals which are projected onto the same feature share a distribution over rewrite rules. That is, for all nonterminals $A, B \in N$ and rules $r \in R$,

$$\phi(A) = \phi(B) \implies p_A(r) = p_B(r). \tag{8.1}$$

The sharing of probability distributions does improve the learnability of the grammars as shown in the computational experiments presented in the following chapters.

8.1 Grammar models of harmony

Before describing the particular grammars, the representation of chord symbols such as Dm^7, G^7, and Cm is introduced. These chords are formalized as pairs consisting of a chord form and a chord root. The *form* of the chord, such as minor seventh □m^7, dominant seventh □^{7}, and minor □m, describes its interval structure relative to the root.[2] The (finite) set of all chord forms is denoted FORMS. The *root* of the chord, such as D, G, or C, is specified in one of two ways. To represent roots that distinguish spellings such that C♯m^7 ≢ D♭m^7, *Tonal Pitch Classes* (TPCs) are used which consist of an integer from zero to seven representing the diatonic scale (C ≡ 0, D ≡ 1,...), plus an additional feature representing alterations such as ♯, ♭, ♭♭, or ♮ (Temperley, 2000). In one of the grammars below, *Pitch Classes* (PCs) are used, a representation on the chromatic scale which does not distinguish spellings. In the following, pitch classes are always denoted in square brackets (C ≡ [0], C♯ ≡ [1] ≡ D♭,...). The combination of a chord form and a root specification using a (tonal) pitch class is called a *(T)PC-chord symbol.* All observed chords and all terminal symbols of harmony grammars are modeled as TPC-chord symbols in the computational experiments.

[2] The square □ stands for an arbitrary root.

8.1.1 TPC-chord grammar

The first grammar type, the TPC-chord grammar, is the simplest. It is a standard context-free grammar which identifies both terminals and nonterminals as TPC-chord symbols. The set of terminals is exactly the set of TPC-chord symbols. The set of nonterminals consists of the set of TPC-chord symbols plus a distinguished start symbol. The set of rules consists of a start rule Start $\longrightarrow A$ for each TPC-chord symbol $A \in N$, a terminal rule $A \longrightarrow \underline{A}$ for each TPC-chord symbol A where the underline indicates a terminal, and all possible duplication rules and headed rules. The chord sequence G^7 $\mathrm{C}^{\triangle}$ could, for example, be derived by:

$$\begin{aligned}
\text{Start} &\longrightarrow \mathrm{C}^{\triangle} \\
&\longrightarrow \mathrm{G}^7 \quad \mathrm{C}^{\triangle} \\
&\longrightarrow \underline{\mathrm{G}}^7 \quad \mathrm{C}^{\triangle} \\
&\longrightarrow \underline{\mathrm{G}}^7 \quad \underline{\mathrm{C}}^{\triangle}
\end{aligned}$$

The feature projection of the TPC-chord grammar is the identity,

$$\phi\colon N \to N, \qquad \phi(A) = A. \tag{8.2}$$

8.1.2 PC-chord grammar

Music theory describes the harmonic system of Jazz as chromatic and enharmonic. This is amongst others reflected in the fact that chord symbols are commonly written using enharmonic equivalent spellings if the correct pitch spelling has two or more accidentals. For instance in the lead sheet of the Jazz standard *Blue moon* printed in the *New Real Book Volume III* in the key of E♭ minor, a chord A♭m^7 is prepared by a tritone-substituted dominant A^7 (Sher, 1995). The correct pitch-spelling of the dominant would, however, be B♭♭^{7}. For critics who argue that this is just an example of notational convenience, not a reflection of the properties of the harmonic space, *Blue moon* provides a second and more fundamental example of enharmonic-equivalent pitch spelling — a sequence of dominant-seventh chords descending by half steps, denoted by D♭^{7} C^7 B^7 B♭^{7}. A correct pitch-spelling of the sequences would be E♭♭♭^{7} D♭♭^{7} C♭^{7} B♭^{7}, highlighting the stepwise descent. However, this spelling does not account for the fact that Db7 can also act as a dominant in Eb major — the ♭VII chord known as the backdoor dominant (e.g., Granroth-Wilding and Steedman, 2014). This example thus highlights the importance of pitch spelling, but also the limitations and ambiguity of chord spellings as used in Jazz lead sheets.

To model the enharmonic equivalence of the harmonic system of Jazz, this study proposes the second kind of harmony grammar, the *PC-chord grammar*, which ignores spelling distinctions in nonterminal chords. In addition to modeling enharmonic equivalence, this allows also a simple kind of transpositional invariance in the rule system.

The set of terminals of a PC-chord grammar is the set of TPC-chord symbols. The set of nonterminals consists of the start symbol and all PC-chord symbols. A PC-chord with root X and chord form $\mathrm{F} \in \textsc{Forms}$ is denoted by X_F. The set of rules consists of a start rule $\text{Start} \longrightarrow X_\mathrm{F}$ for each PC-chord symbol X_F, a terminal rule for each distinct *spelling* of a PC into a corresponding TPC (e.g., $[1] \mapsto \mathrm{C}\sharp$ and $[1] \mapsto \mathrm{D}\flat$), a duplication rule $\textsc{dupl} : N \twoheadrightarrow N^2$ with $\mathrm{dom}(\textsc{dupl}) = N \setminus \{\text{Start}\}$, and the following headed rules where $\iota \in \{[0], \dots, [11]\}$ denotes an interval on the chromatic scale and the root Y equals $X + \iota \bmod 12$:

$$\textsc{left}_{\iota,\mathrm{F'}}(X_\mathrm{F}) = X_\mathrm{F} \;\; Y_{\mathrm{F'}} \qquad \mathrm{dom}(\textsc{left}_{\iota,\mathrm{F'}}) = \{X_\mathrm{F} \mid X_\mathrm{F} \neq Y_{\mathrm{F'}}\} \tag{8.3}$$

$$\textsc{right}_{\iota,\mathrm{F'}}(X_\mathrm{F}) = Y_{\mathrm{F'}} \;\; X_\mathrm{F} \qquad \mathrm{dom}(\textsc{right}_{\iota,\mathrm{F'}}) = \{X_\mathrm{F} \mid X_\mathrm{F} \neq Y_{\mathrm{F'}}\} \tag{8.4}$$

These rules elaborate the chord X_F by chords with chord form F' whose roots are ι semitones higher than X, for example, $\textsc{left}_{9,\mathrm{m}^7}([0]^\triangle) = [0]^\triangle \; [9]^7$ and $\textsc{right}_{7,\mathrm{m}^7}([7]^7) = [2]^7 \; [7]^7$. The domain restrictions ensure that duplication rules and headed rules do not overlap. The chord sequence $\mathrm{G}^7 \; \mathrm{C}^\triangle$ could for example be derived by:

$$\begin{aligned} \text{Start} &\longrightarrow [0]^\triangle \\ &\longrightarrow [7]^7 \quad [0]^\triangle \\ &\longrightarrow \mathrm{G}^7 \quad [0]^\triangle \\ &\longrightarrow \mathrm{G}^7 \quad \mathrm{C}^\triangle \end{aligned}$$

On one hand, the representation of chord roots as PCs instead of TPCs loses information, for instance, about the key. Consider for example the dominant-seventh chords $\mathrm{B}^7$and $\mathrm{C}\flat^7$; they both have the pitch class 11 as root, but convey different information about the key in which they appear. The first chord $\mathrm{B}^7$is the fifth scale degree in the keys E major and E minor and is thus likely to appear in these keys. The second chord $\mathrm{C}\flat^7$would be the scale degree ♭♭VI in the key E major and ♭VI in E minor. It is therefore not likely to appear in these keys. Instead, it is for example more likely to appear as tritone-substituted dominant in the key B♭major — as scale degree ♭II — or as tritone-substituted double dominant in the key E♭major — as scale degree ♭II/V.

On the other hand, the representation of chord roots as PCs enables a simple transpositionally invariant parametrization of the grammar where all chords of the same form share a rewrite distribution. This is accomplished using the feature projection

$$\phi \colon N \to \textsc{Forms} \uplus \{\text{Start}\} \tag{8.5}$$

that maps the start symbol to itself, $\phi(\text{Start}) = \text{Start}$, and a PC-chord to its chord form, $\phi(X_\mathrm{F}) = \mathrm{F}$. The probability of rewriting a PC-chord X_F with a rule $r \in R$ is therefore given by

$$p_{X_\mathrm{F}}(r) = p_{\phi(X_\mathrm{F})}(r) = p_F(r). \tag{8.6}$$

This parametrization makes the PC-chord grammar robust to different keys of chord sequences and properly handles local modulations.

8.1.3 Unsupervised harmony grammar with induced nonterminal categories

Fully unsupervised grammar induction is the task to infer the nonterminals, the rules, and the rewrite probabilities of a grammar from the observation of only sequential data. Since the TPC-chord and the PC-chord grammars use predefined nonterminal representations, grammar induction using these models could be considered to be weakly supervised. To investigate which nonterminal representations are learnable from a dataset of Jazz chord sequences, a grammar model with induced nonterminal categories is introduced in the following.

This grammar uses $M \in \mathbb{N}_+$ unstructured nonterminals; it is a standard context-free grammar whose set of nonterminals comprises the start symbols and the natural numbers from 1 to M. The set of rules consists of a start rule Start $\longrightarrow m$ for each $m \in \{1, \dots, M\}$, a terminal rule $m \longrightarrow a$ for each combination of $m \in \{1, \dots, M\}$ and TPC-chords $a \in T$, and all possible duplication rules and headed rules. The feature projection of the unsupervised harmony grammar is the identity. The chord sequence G^7 $\mathrm{C}^{\triangle}$ could for example be derived as shown below. Note that the nonterminal numbers are completely arbitrary in this grammar and in this example. Their meaning arises in the way that the model learns to use these symbols to (re-)generate observed data. This is why the nonterminals of the unsupervised grammar are called *induced categories*.

$$\begin{aligned} \text{Start} &\longrightarrow 4 \\ &\longrightarrow 1 \quad 4 \\ &\longrightarrow \mathrm{G}^7 \quad 4 \\ &\longrightarrow \mathrm{G}^7 \quad \mathrm{C}^{\triangle} \end{aligned}$$

8.2 A joint grammar model of rhythm

To improve the learning of the harmony grammars, all models of harmonic structure are paired with a model of rhythmic structure using the product construction described in Section 5.5. The feature projection of a product grammar is thereby defined as the function that maps a pair of nonterminals component-wise to the pair of their features. The proposition of product-grammar models is a core contribution of this thesis. The PACFG framework allows to formalize product grammars in the same way as individual grammars, allowing for simple model architectures.

The rhythm grammar was already motivated and introduced in Section 5.1; this paragraph gives a brief summary. The rhythm grammar uses rational numbers $0 < u \leq 1$ as terminals that represent chord durations relative to the entire chord sequence. The set of nonterminals $N \cong T \uplus \{\text{Start}\}$ establishes a one-to-one correspondence between terminals and nonterminals, analogous to the TPC-chord grammar. The set of rules consists of one start rule Start $\longrightarrow$ 1, one terminal rule that maps nonterminals to their respective terminals, and binary split rules

$$\textsc{split}_s \colon N \setminus \{\text{Start}\} \to N^2, \qquad \textsc{split}_s(u) = (su) \quad (u - su) \tag{8.7}$$

for finitely many split ratios $s \in \mathbb{Q}, 0 < s < 1$. To parameterize the rewrite probabilities independently from the chord durations, the feature projection

$$\phi \colon N \to \{0,1\}, \qquad \phi(A) = \mathbb{1}(A = \text{Start}) \tag{8.8}$$

is used that checks whether a nonterminal is the start symbol or not. The rewrite probabilities therefore do not depend on the particular duration of a chord somewhere in a derivation, but only on the split ratios of the rules that will be applied to the chord.

To illustrate the generative mechanics of product grammars, the following example shows a derivation of the chord sequence $\text{G}^7\ \text{C}^{\triangle}$ with durations $1/2$ and $1/2$ using the product of the PC-chord grammar with the rhythm grammar. The product rules (e.g., pairs of rules from the component grammars) used in the derivation steps are indicated as subscripts on the derivation arrows.

$$\begin{aligned}
\text{Start} &\longrightarrow_{(\textsc{start}_{[0]^{\triangle}},\textsc{start})} && ([0]^{\triangle}, 1) \\
&\longrightarrow_{(\textsc{right}_{7,\square^7},\textsc{split}_{1/2})} && ([7]^7\ , 1/2) \quad ([0]^{\triangle}, 1/2) \\
&\longrightarrow_{(\textsc{terminate}_\flat,\textsc{terminate})} && (\text{G}^7\ \ , 1/2) \quad ([0]^{\triangle}, 1/2) \\
&\longrightarrow_{(\textsc{terminate}_\flat,\textsc{terminate})} && (\text{G}^7\ \ , 1/2) \quad (\text{C}^{\triangle}\ \ , 1/2)
\end{aligned}$$

Here, the rule $\textsc{terminate}_\flat$ denotes the terminal rule of the PC-chord grammar that corresponds to the pitch-spelling which uses flats. However, it would be the same if the spelling which uses sharps would have been used.

9 Supervised grammar learning[1]

This chapter presents computational experiments in which the rewrite probabilities of harmony grammars are learned from the expert-created tree analysis of the Jazz Harmony Treebank (JHT, see Chapter 4). The rule sets of the grammars include all possible headed rules and duplication rules. Prior to learning from the treebank, all rewrite probabilities are considered equal. The large amount of rewrite rules both smooths the grammar models and makes them robust against chord symbols not seen during training, at the cost of increasing parsing time. However, the parsing speed of the models were not a problem in practice. All models ran (including training and prediction, excluding plotting and bootstrapping) in under 20 seconds on a MacBook Pro (15-inch, 2019) with a 2.6 GHz 6-Core Intel Core i7 processor and 16GB memory.[2]

The basic question studied in the supervised experiments is: which grammar models generalize best from the observation of example tree analyses. These experiments are called supervised, because the observed tree analyses guide the learning of the models. In contrast, the next chapter presents unsupervised experiments in which harmony grammars are learned from the observation of the chord sequences alone (without observing any tree analyses). The function of the supervised experiments is therefore also to set expectations for the unsupervised experiments.

The TPC-chord grammar model (transpositionally dependent parameterization) and the PC-chord grammar model (transpositionally independent parameterization) are tested with and without a joint grammar model of rhythm, see Chapter 8 for more details. The results of the experiments are analysed quantitatively and qualitatively to study the following two main hypotheses:

[1] Parts of the results presented in this chapter are already published in a peer-reviewed article: Harasim, D., O'Donnell, T. J., and Rohrmeier, M. (2019). Harmonic Syntax in Time: Rhythm Improves Grammatical Models of Harmony. *Proceedings of the 20th International Society for Music Information Retrieval Conference*

[2] All models are implemented in Haskell. The code is planed to be published in the form of a grammar inference library in the future.

1. Jointly modeling rhythm improves grammar models of harmony.
2. A transpositionally invariant parameterization improves garmmar models of harmony.

Jointly modeling rhythm is likely to improve grammar models for Jazz harmony, because of the rhythmic regularity of harmonic phrases described in Sections 1.5 and 5.1. A transpositionally invariant parameterization of harmonic rewrite rules is likely to improve harmonic grammar models, because of human's relative pitch perception. Therefore, it makes sense to assume a learning mechanism which focuses on the relation between chords instead of absolute pitches. This key invariance of harmonic relations is applicable to Jazz but has been questioned for other styles such as Western classical music (Quinn and White, 2017). Furthermore, the usage of transpositional invariant rules decreases the size of the grammar, which could either be beneficial because it increases the learning speed, or disadvantageous if too much information is lost.

The experiments provide strong evidence for both hypotheses. In fact, jointly modeling rhythm improves the (unlabeled) accuracy of the predicted derivations (defined below in detail) by about 15%. In contrast, the effect size of transpositional invariance is much lower. Transpositional invariance improves the accuracy of the predictions by about 2%. Additional findings and insights are presented and discussed later with the results.

Supervised learning of rewrite probabilities by observation of derivation trees is much easier than unsupervised learning from the observation of bare chord sequences. Such supervised learning can be performed by simple counting of the rewrite rules used in the derivations; it does not require variational Bayesian approximations. Using the notation developed and the equations derived in Chapter 7, the probability of a grammar's rewrite probabilities $\boldsymbol{\theta}$ conditioned on observed derivations $\bar{\boldsymbol{r}}$ is given in closed form as a product of Dirichlet distributions (see Equation 7.54),

$$p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}}) = \prod_{f \in \Phi} \frac{1}{\mathrm{B}(\hat{\boldsymbol{\alpha}}^f)} \prod_{r \in R_f} (\hat{\boldsymbol{\theta}}_r^f)^{\hat{\boldsymbol{\alpha}}_r^f - 1}, \tag{9.1}$$

where

$$\hat{\boldsymbol{\alpha}}_r^f = \boldsymbol{\alpha}_r^f + \sum_{i=1}^{I} \sum_{k=1}^{|\boldsymbol{r}^i|} \mathbb{1}\left(f = \phi(A_k^i)\right) \mathbb{1}\left(r = \boldsymbol{r}_k^i\right) \tag{9.2}$$

for features $f \in \Phi$ and rules $r \in R_f$. The notation is chosen such that r denotes a rule, $\boldsymbol{r}$ denotes a sequence of rules that represent a derivation tree, $\bar{\boldsymbol{r}}$ denotes a dataset of derivations, $\boldsymbol{r}^i$ denotes the i-th derivation in the dataset $\bar{\boldsymbol{r}}$, and $\boldsymbol{r}_k^i$ denotes the k-th rule of the i-th derivation. The form of $p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}})$ results from the fact that the rewrite-rule distributions are sampled independently from Dirichlet distributions for each feature. For the experiments, all hyperparameters α_r^f are set to 0.1. The resulting prior distribution $p(\boldsymbol{\theta})$ does not favor any particular rewrite rule, but encodes that the rewrite-rule distributions are expected to have rather low entropy for all features.[3] Furthermore, the learning of product grammars is performed by propagating the rule counts to the rule distributions of the component grammars.

The treebank analyses which apply open constituents are used for supervised learning. The asterisks that mark the roots of open constituents as introduced in Chaper 4 are, however, not used in order to enhance the compatibility with the unsupervised grammar models in the next chapter.

9.1 Quantitative evaluation measures

All model evaluation is performed by *leave-one-out cross-validation*. That is, each model is trained (by rule counting) on 149 derivation trees of the JHT and evaluated on the chord sequence of the left-out tune. Leave-one-out cross-validation thus has the advantage that the models can be trained and evaluated on all data without evaluating any model on a tune that it was trained on. It therefore avoids overfitting. Leave-one-out cross-validation is of particular importance for the presented experiments, because the JHT is too small to be split in fixed subsets for training and evaluation.

9.1.1 Tree prediction assessment

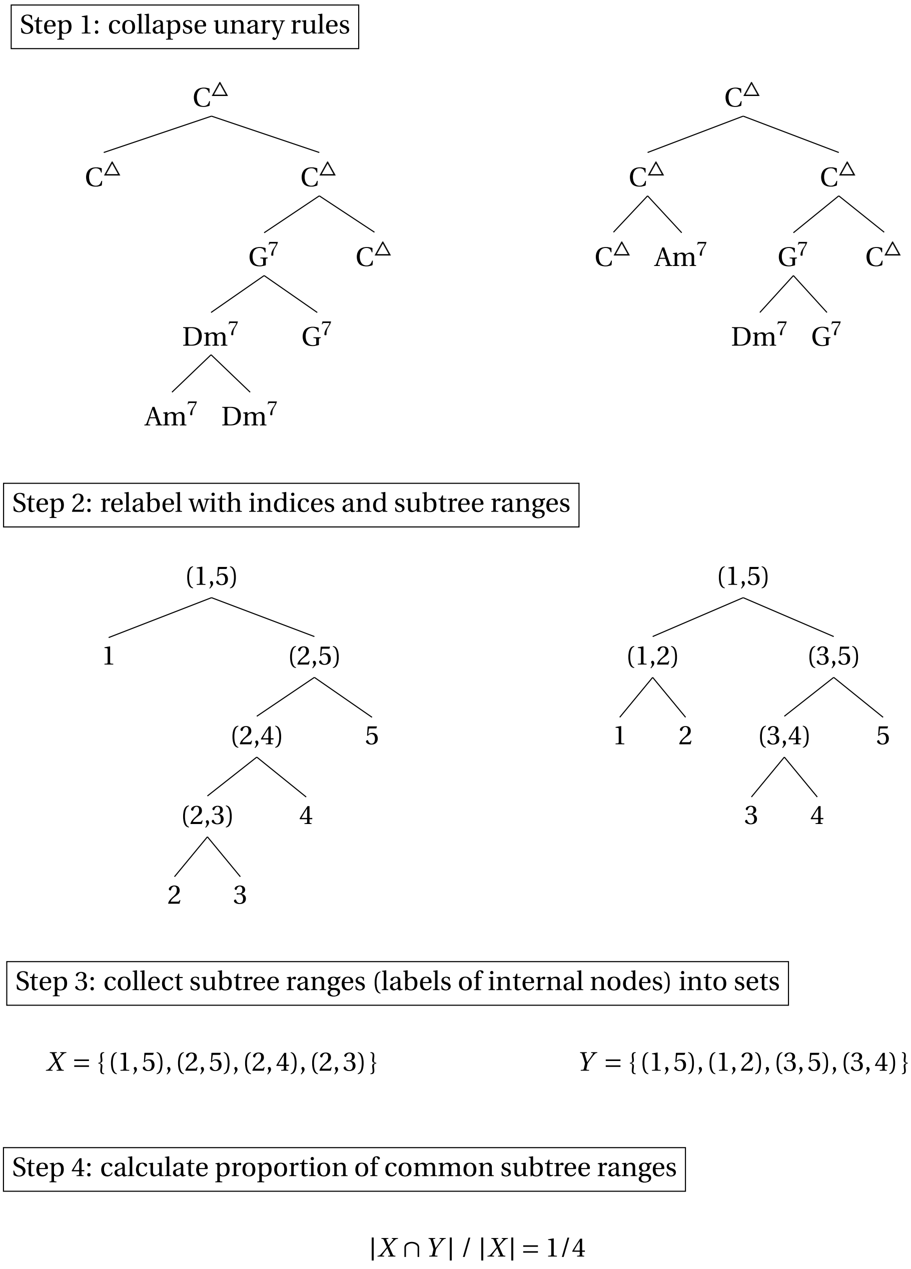
The trained grammars are used to predict derivation trees for chord sequences. The tree prediction is defined as the most probable derivation of the respective chord sequence. The most probable derivation (also called maximum a posteriori estimation) of a sequence $\boldsymbol{w} \in T^*$ is the derivation $\boldsymbol{r} \in \textsc{Der}(\boldsymbol{w})$ which maximizes the probability $p(\boldsymbol{r} \mid \bar{\boldsymbol{r}})$, where $\bar{\boldsymbol{r}}$ denotes the dataset of observed trees during training. It is given by

$$\operatorname*{argmax}_{\boldsymbol{r} \in \textsc{Der}(\boldsymbol{w})} p(\boldsymbol{r} \mid \bar{\boldsymbol{r}}) = \operatorname*{argmax}_{\boldsymbol{r} \in \textsc{Der}(\boldsymbol{w})} \int_{\boldsymbol{\theta}} p(\boldsymbol{r} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}}) \tag{9.3}$$

and can be computed in the semiring parsing framework as described in Section 6.2 near Equation 6.20.

The tree predictions of the models are evaluated using two quantitative measures which assess the similarity between the predictions and the JHT trees. Both measures yield rational numbers between 0 and 1, where 0 stands for minimal and 1 for maximal similarity. Since the aim is to also evaluate models which use different nonterminals than used in the JHT (e.g., nonterminals that are not TPC chords), nonterminal-agnostic measures are chosen. Such measures are called *unlabeled* similarity measures. In the following, the term *accuracy* is used for a similarity measure which compares a tree prediction to a JHT tree.

[3] The *(Shannon) entropy* of a discrete random variable's distribution $p(x)$ is defined as $-\sum_{x \in X} p(x) \log p(x)$, where X denotes the set of all possible values the random variable can take. It can be interpreted as a measure for how unpredictable the distribution $p(x)$ is. See for instance MacKay (2003) for more explanation.


Step 1: collapse unary rules
C△
C△
C△
G7
C△
Dm7
G7
Am7
Dm7
C△
C△
C△
C△
Am7
G7
C△
Dm7
G7
Step 2: relabel with indices and subtree ranges
(1,5)
1
(2,5)
(2,4)
5
(2,3)
4
2
3
(1,5)
(1,2)
(3,5)
1
2
(3,4)
5
3
4
Step 3: collect subtree ranges (labels of internal nodes) into sets
Step 4: calculate proportion of common subtree ranges

$$X = \{(1,5),(2,5),(2,4),(2,3)\} \qquad Y = \{(1,5),(1,2),(3,5),(3,4)\}$$

$$|X \cap Y| \;/\; |X| = 1/4$$

Figure 9.1 – (Unlabeled) tree accuracy calculation in 4 steps. The two columns represent the simultaneous computations for the two input trees. See the main text for more details.

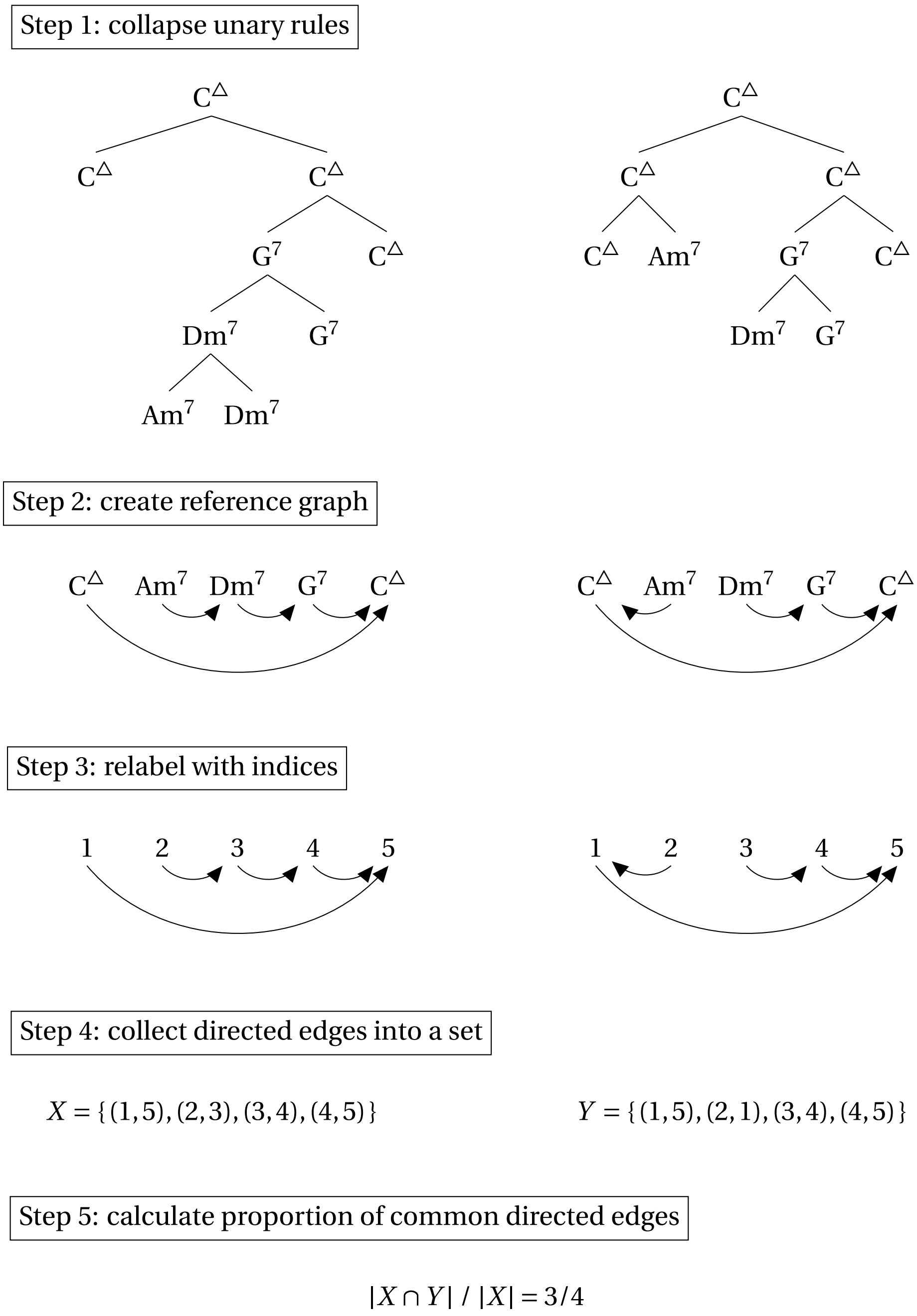
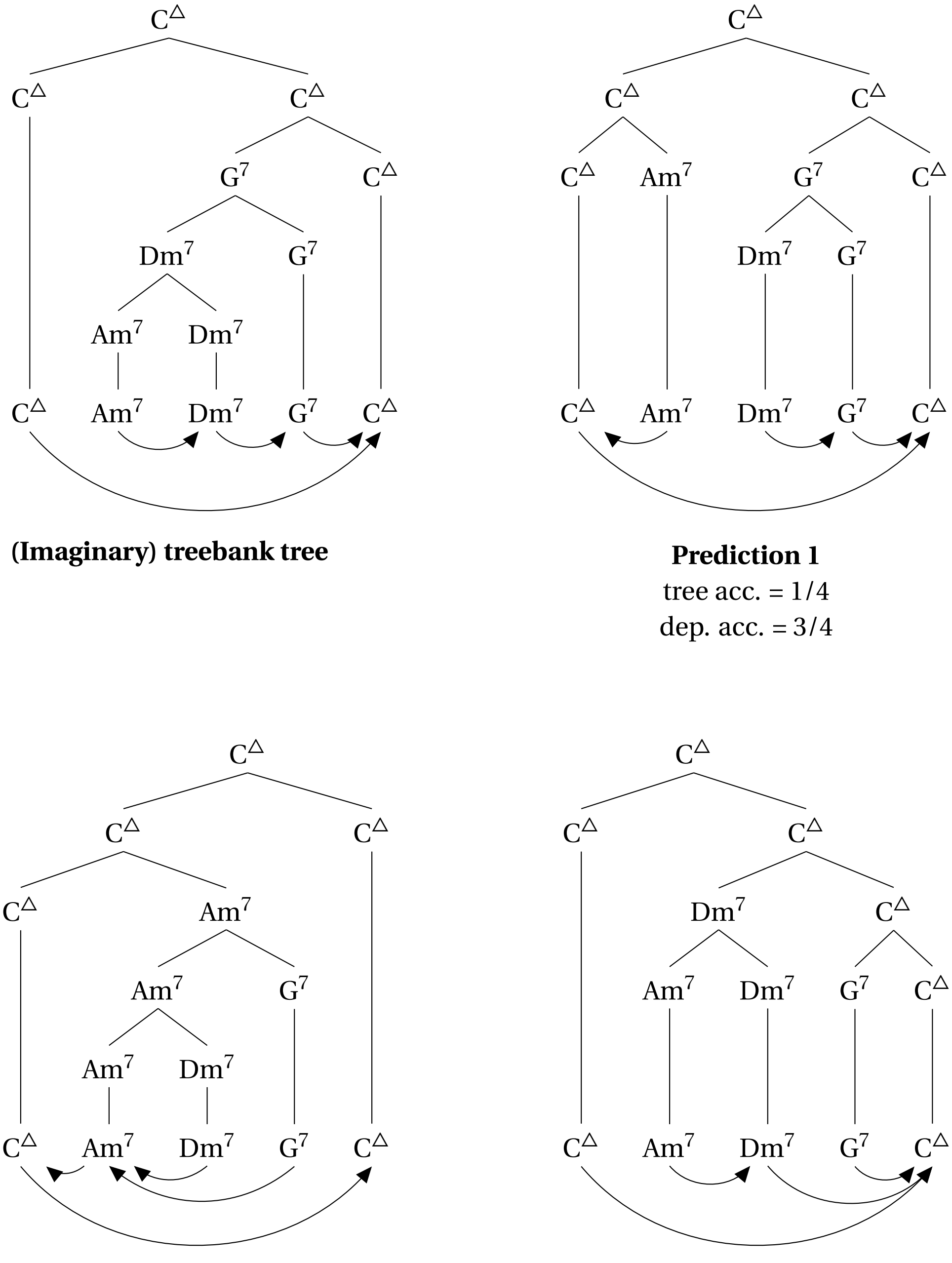

Step 1: collapse unary rules
C△
C△
C△
G7
C△
Dm7
G7
Am7
Dm7
C△
C△
C△
C△
Am7
G7
C△
Dm7
G7
Step 2: create reference graph
C△ Am7 Dm7 G7 C△
C△ Am7 Dm7 G7 C△
Step 3: relabel with indices
1 2 3 4 5
1 2 3 4 5
Step 4: collect directed edges into a set
X = {(1,5),(2,3),(3,4),(4,5)}
Y = {(1,5),(2,1),(3,4),(4,5)}
Step 5: calculate proportion of common directed edges
|X ∩ Y| / |X| = 3/4

Figure 9.2 – (Unlabeled) dependency accuracy calculation in 5 steps. The two columns represent the simultaneous computations for the two input trees. See the main text for more details.

The first measure is called *(unlabeled) tree accuracy*; it is defined as the proportion of correctly predicted constituents that comprise at least two symbols. In this definition, a constituent refers to a subsequence whose chord symbols are the leafs of a subtree. A constituent can therefore be represented by its start and end index in the full chord sequence that describe its *range*. The following presents a definition that is well-defined for all derivation trees constructed from unary and binary rules. The tree-accuracy calculation is visualized in Figure 9.1 in 4 steps. The first step collapses the unary rule applications of the trees to be compared, yielding two binary trees. Unary rule applications are irrelevant to the proposed measure because only constituents that comprise at least two terminals are considered and nonterminal labels are ignored. The second step relabels the leafs (the elements of the chord sequence) with their sequence indices and the internal tree nodes with the index range they span over. Such a range is defined as a pair of two indices, the index of the leftmost and the index of the rightmost leaf. Since each internal tree node uniquely represents the subtree whose root it is, the range of an internal tree node can also be considered the range of the corresponding subtree and its corresponding constituent. The third step collects the ranges of the trees into two sets and the final step yields the tree accuracy as the proportion of common subtree ranges.

The second measure is called *(unlabeled) dependency accuracy*; it is defined as the proportion of correctly predicted harmonic references. Dependency accuracy is, like tree accuracy, an evaluation measure used in computational linguistics. This paragraph presents a definition of dependency accuracy that is well-defined for derivation trees constructed from duplication rules, headed rules, and unary start and terminal rules. The dependency accuracy calculation is visualized in Figure 9.2 in 5 steps. The first step is equivalent to the tree-accuracy calculation; it collapses the unary rule applications. The second step transforms the tree into its corresponding reference graph by mapping left-headed rule application to leftwards-directed reference arrows and right-headed as well as duplication rule applications to rightwards-directed reference arrows (see Chapter 1 for more details). The third step relabels the vertices of the reference graph (the chord symbols) with the indices of the sequence. The fourth step then collects the edges of the resulting graphs (represented by ordered pairs of sequence indices) into two sets. The order of the pair thereby represents the direction of the arrow. The fifth and last step finally yields the dependency accuracy as the proportion of common chord references.

Tree accuracy and dependency accuracy are correlated, but not deterministically dependent. In some cases the former is higher, in other cases the latter. However, dependency accuracy is in practice usually higher than tree accuracy, because it punishes a wrong decision only once. Figure 9.3 shows trees and reference graphs for an imaginary treebank tree and three tree predictions to illustrate the relation between tree accuracy and dependency accuracy. The chord sequence derived by all trees is the exemplary cadential progression $\mathrm{C}^{\triangle}$ Am^7 Dm^7 G^7 $\mathrm{C}^{\triangle}$. The first tree prediction shows an example in which the wrong attachment of the chord Am^7 is punished 3 times by the tree accuracy measure, but only once by dependency accuracy. The word *wrong* is here of course understood in relation to the treebank, not as a normative



(Imaginary) treebank tree

Prediction 1
tree acc. = 1/4
dep. acc. = 3/4

Prediction 2
tree acc. = 3/4
dep. acc. = 1/4

Prediction 3
tree acc. = 3/4
dep. acc. = 3/4

Figure 9.3 – Comparison of tree accuracy and dependency accuracy for three examples.

term. The second example is a derivation tree which is in its unlabeled structure similar to the treebank tree, but whose labels introduce a quiet different reference graph. This example is complementary to the first example in the sense that the tree accuracy is here three times higher than the dependency accuracy. The third and last example shows a tree prediction in which the Dm^{7}chord does not refer to the G^{7}chord, but directly to C$^\triangle$. Both tree accuracy and dependency accuracy yield the same value of 3/4 in this case.

9.1.2 Treebank-independent evaluation

Additional to the accuracy evaluation of the tree predictions, which depends on the treebank analyses, all models are also evaluated with a treebank-independent measure. Note that the training of the models still depends on the treebank, only the evaluation of a trained model is treebank-independent. An evaluation measure based on the predictive probabilities of the chord sequences is used,

$$p(\boldsymbol{w} \mid \bar{\boldsymbol{r}}) = \sum_{\boldsymbol{r} \in \mathrm{DER}(\boldsymbol{w})} p(\boldsymbol{r} \mid \bar{\boldsymbol{r}}) = \sum_{\boldsymbol{r} \in \mathrm{DER}(\boldsymbol{w})} \int_{\boldsymbol{\theta} \in \Theta} p(\boldsymbol{r} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \bar{\boldsymbol{r}}), \tag{9.4}$$

where $\bar{\boldsymbol{r}}$ denotes the dataset of observed trees during training and $\boldsymbol{w} \in T^+$ denotes the predicted sequence of terminals. The probability of all $I \in \mathbb{N}$ sequences $\bar{\boldsymbol{w}}$ is the product

$$p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}) = \prod_{i=1}^{I} p(\boldsymbol{w}^i \mid \bar{\boldsymbol{r}}^{\neg i}), \tag{9.5}$$

where $\bar{\boldsymbol{r}}^{\neg i}$ denotes the set of all observed trees except for the i-th one. Note that in this equation, a derivation $\boldsymbol{r}^i$ is the treebank derivation of the sequence $\boldsymbol{w}^i$, but the product is calculated using leave-one-out cross-validation as described above.

The so called *Mean Log Predictive* (MLP) of the sequence dataset $\bar{\boldsymbol{w}}$ is used as treebank-independent evaluation measure,

$$\mathrm{MLP}(\bar{\boldsymbol{w}}) = \frac{1}{I} \sum_{i=1}^{I} \log p(\boldsymbol{w}^i \mid \bar{\boldsymbol{r}}^{\neg i}) \tag{9.6}$$

where log denotes the natural logarithm. The MLP is only reported for single-component grammars to avoid the explicit calculation of normalizing constants for product grammars (see Section 5.5). One property of the MLP is that it equals the logarithm of the probability of the dataset divided by its size,

$$\frac{1}{I} \sum_{i=1}^{I} \log p(\boldsymbol{w}^i \mid \bar{\boldsymbol{r}}) = \frac{1}{I} \log \prod_{i=1}^{I} p(\boldsymbol{w}^i \mid \bar{\boldsymbol{r}}) = \frac{1}{I} \log p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}). \tag{9.7}$$

It is therefore a measure of the data's probability per sequence. Logarithms of the probabilities are used to improve their readability, because the sequence probabilities are very small.

Additional to reporting the MLPs, Bayes factors are reported for pairwise model comparison. The term grammar model describes an ACFG with a parameterization and a prior distribution over parameters. In this understanding, a grammar model is not a particular PACFG, but characterized by a family of PACFGs of which the parameters might be inferred from datasets. Given two grammars models M_1 and M_2 with the same set of terminals, the *Bayes factor* is defined as

$$\mathrm{BF}(M_1, M_2) = \frac{p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}, M_1)}{p(\bar{\boldsymbol{w}} \mid \bar{\boldsymbol{r}}, M_2)}, \tag{9.8}$$

again computed via cross-validation. Under the assumption that both grammar models are equally probable a priori, their Bayes factor is equal to the posterior odds in favor of the model M_1. That is, the Bayes factor indicates how much better model M_1 fits to the data than model M_2. According to Kass and Raftery (1995), Bayes factors from 20 to 150 provide strong evidence and Bayes factors from over 150 very strong evidence for the model M_1.

The advantages of using Bayes factors instead of using frequentist methods such as classical significance tests are numerous (Halsey et al., 2015; Wagenmakers et al., 2018; Ho et al., 2019). The first advantage is that is calculates directly what scientists are interested in, since the goal of scientific investigation is usually to quantify the evidence *for* a theory or a model, not against it. The interpretation of Bayes factors is therefore straight-forward. In contrast, p-values of frequentist hypothesis tests are hard to interpret correctly and have a high risk of being misunderstood, for example as posterior probabilities that the null hypothesis is correct. Instead, a p-value is defined as the relative frequency of how often the null hypothesis is rejected by mistake. Furthermore, if multiple tests are performed in a study, which is usually the case, then the threshold under which null hypotheses are rejected needs to be adjusted, because the more tests the greater the probability of rejecting a null hypothesis by mistake.

One important drawback of Bayes factors is that they are only interpretable if the prior distributions of the models are closely related (Lavine and Schervish, 1999). This is for example the case if uninformative “flat” priors are used. However, Bayes factors can not be applied to compare models whose priors are used for regularization purposes as in the second unsupervised experiment presented in the next chapter.

9.2 Bayesian bootstrap

The usage of Bayes factors to compare grammar models leaves at least one question unanswered: How much does the Bayes factor depend on the concrete dataset of chord sequences at hand? Theoretically, this question could be studied by 1) repeatedly sampling an equally sized dataset of Jazz standards (150 tunes), 2) derivation-tree analysis of the tunes by experts, and 3) training and evaluation of the models to compute the Bayes factor with respect to the newly sampled dataset. The values created in this manner directly represent the uncertainty of the Bayes factor with respect to the dataset. Since the described procedure is, however, not feasible in practice, the Bayesian bootstrap is used to approximate it as described by Rubin (1981).

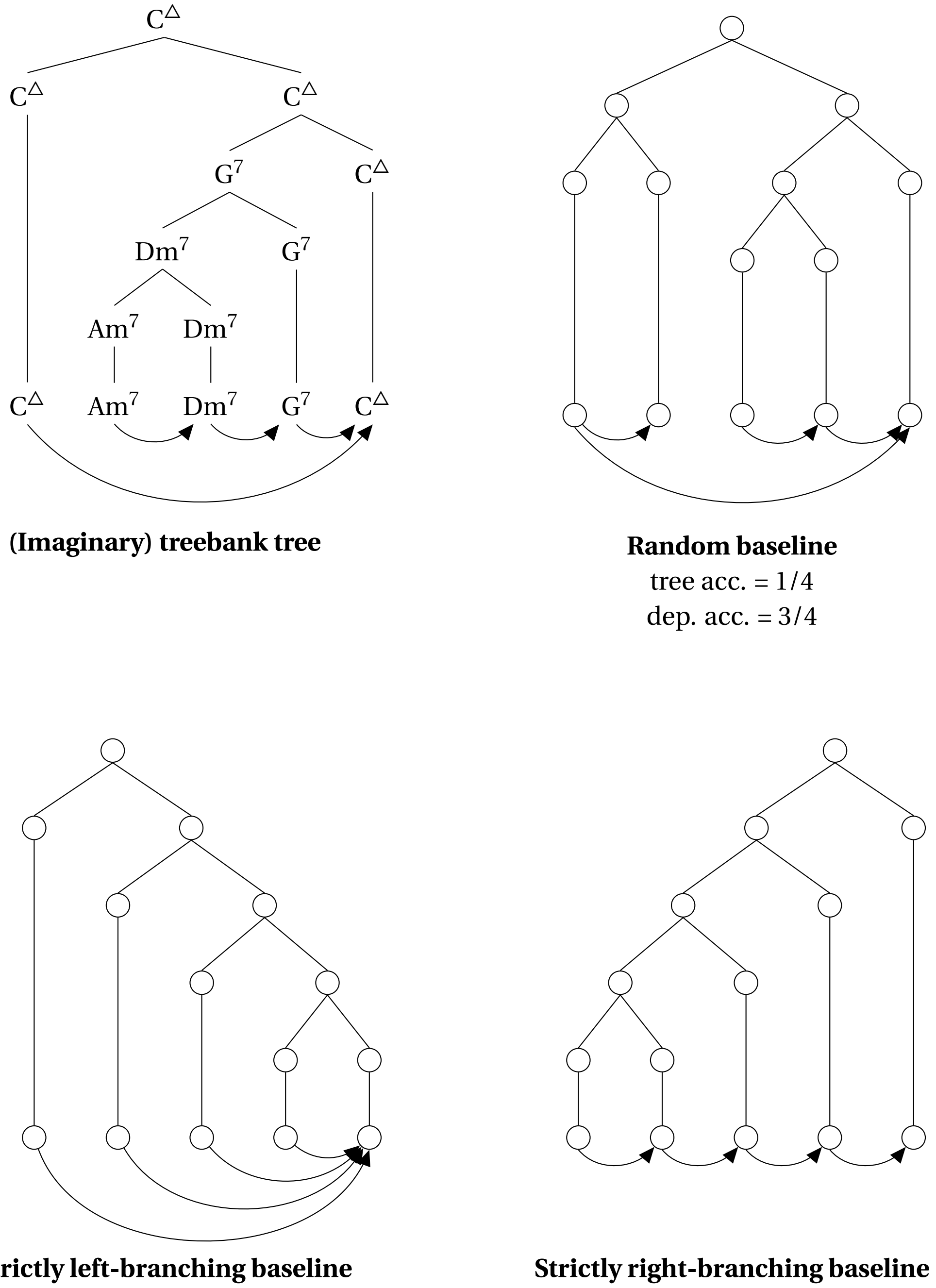
As in the theoretical scenario, the Bayesian bootstrap considers the dataset a random variable. The Bayes factor is consequently also a random variable from which samples can be obtained in order to determine its uncertainty with respect to the dataset. The Bayesian bootstrap approximates the sampling of new datasets by resampling the one and only dataset given. The resampling is done by the following generative procedure in two steps. First, a vector of weights is sampled from a symmetric Dirichlet distribution with a concentration parameter of 2 (this choice is explained in the next paragraph). The length of the weight vector equals the size of the dataset; this vector thus assigns a positive weight to every element of the dataset and all weights sum up to 1. In the second step, the dataset is resampled by sampling from a multinomial distribution according to the weight vector sampled in the first step.

The choice of the symmetric Dirichlet distribution with a concentration parameter of 2 in the first step is explained as follows. A priori, each weight vector is modeled equally probable. The prior distribution over weight vectors is thus a symmetric Dirichlet distribution with a concentration parameter of 1. The observation of each element of the dataset (each tune) exactly once implies that the posterior distribution is a symmetric Dirichlet distribution with a concentration parameter of 2. This follows directly from the fact that the Dirichlet distribution is a conjugate prior for the multinomial distribution (see Equation 7.26).

The Bayesian bootstrap is closely related to the nonparametric bootstrap used in frequentist statistics where the resampling is performed as simple sampling with replacement. The crucial difference is not how the resampling is performed, but for what it is used and how it is interpreted. While in frequentist statistics the boostrap is commonly used for a test statistic, the Bayesian bootstrap is applied to sample from the posterior distribution of a random variable.

Note that there is a conflict when the bootstrap is jointly used with leave-one-out cross-validation. When the latter is applied to a resampled dataset in order to train a model and evaluate it on a left-out data point, then the model is likely to be evaluated on data it was trained on because resampling is performed with replacement. In this study, leave-one-out cross-validation is therefore applied to the original dataset to obtain a set of evaluation measures per chord sequence (e.g., sequence probabilities or tree accuracies). That set of evaluation measures is then resampled to estimate the uncertainty associated with the dataset's random variable.

By applying the Bayesian bootstrap for estimating the uncertainty of the Bayes factors, each chord sequence of the musical idiom of Jazz standards considered in this study is assumed to be similar to a chord sequence of the treebank. The results are therefore at least valid with respect to the idiom that the treebank represents.



(Imaginary) treebank tree

Random baseline
tree acc. = 1/4
dep. acc. = 3/4

Strictly left-branching baseline
tree acc. = 1/4
dep. acc. = 2/4

Strictly right-branching baseline
tree acc. = 1/4
dep. acc. = 3/4

Figure 9.4 – Baseline examples in comparison to one imaginary treebank tree. All baseline predictions are unlabeled and strictly right-headed.

9.3 Baselines

The grammar models are compared to three baselines: a strictly left-branching, a strictly right-branching, and a random baseline. Figure 9.4 shows one example tree for each baseline in comparison to an exemplary treebank tree. Each rule application of every baseline is furthermore assumed to be right-headed. This strict right-headedness is assumed, because most of the rule applications in the JHT are right-headed. For the strictly left-branching baseline, strict right-headedness implies a reference structure in which each chord is directly referencing the last chord of the sequence. For the strictly right-branching baseline, it implies that each chord is referencing its successor. The reference structure of the strictly right-branching baseline is therefore equivalent to the reference structure of a bigram model. The tree prediction of the random baseline samples a binary tree uniformly at random from the set of all binary trees which have as many leafs as the sequence that is to be predicted has chord symbols.

Note that all baselines model only unlabeled tree and reference structures. Therefore, they serve as baselines for tree accuracy and dependency accuracy, but not for MLP.

9.4 Quantitative results and discussion

The results of the supervised-grammar-learning experiments are summarized in Table 9.1. The averages of the evaluation measures tree accuracy and dependency accuracy are reported along with the mean log predictive for the grammar models and three baselines. The average tree heights are additionally reported to assess the balancedness of the tree predictions. The balancedness of a model's predictions can be quantified by the average tree height, because the lengths of the treebank sequences are constant throughout the experiments.

Dependency accuracy is constantly higher as tree accuracy for both product grammars and both single-component grammars for harmony. One explanation for this is that tree accuracy commonly punishes wrong predictions multiple times, as mentioned in Section 9.3. Because of the similarity of the accuracy measures, only one of them, namely tree accuracy, is discussed in detail in the following.

The results confirm both hypotheses described at the beginning of this chapter:

1. Jointly modeling rhythm improves grammar models of harmony.
2. A transpositionally invariant parameterization improves garmmar models of harmony.

Moreover, the effect size of jointly modeling rhythm is higher than of transpositional invariance. The following section first discusses the evidence for the second and then for the first hypothesis.

Model	Tree Acc.	Dep. Acc.	MLP	Tree Height
TPC chord	46.2	57.9	-96.12	12.7
PC chord	**48.1**	**59.8**	**-82.57**	12.4
TPC chord & rhythm	61.7	71.8	–	7.1
PC chord & rhythm	**63.1**	**74.2**	–	7.3
only rhythm	57.7	–	-54.94	6.5
random baseline	19.3	35.6	–	14.2
left-branching baseline	13.7	58.6	–	26.0
right-branching baseline	17.5	4.7	–	26.0

Table 9.1 – Supervised-grammar-learning results. Mean tree accuracy, mean dependency accuracy, mean log predictive, and mean tree height are reported for each model. Higher is better for all evaluation measures. Tree accuracy and dependency accuracy are reported in percent. The tested models are the TPC-chord grammar and the PC-chord grammar, with and without a joint rhythm grammar. For comparison, all applicable evaluation measures are also reported for the single-component rhythm grammar and 3 baselines; the single-component rhythm grammar does not predict dependency relations and the baselines do not predict terminal symbols. Both with and without jointly modeling rhythm, the PC-chord grammar significantly outperforms all other models with respect to all evaluation measures (see Figures 9.5–9.7 for evidence quantification).

9.4.1 Transpositional invariance

Both with and without jointly modeling rhythm, the transpositionally invariant PC-chord grammar outperforms its respective TPC-chord grammar and all baselines with respect to all evaluation measures. The effect of transpositional invariance is, with an improvement of about 2% in both tree and dependency accuracy, subtle but consistent. The question of whether this improvement is significant is studied by taking a closer look at the mean tree-accuracy differences of the PC-chord and the TPC-chord grammar models. The results are analogous for dependency-accuracy differences. The question of significance is a question about how much the measured accuracy difference depends on the considered dataset. In other words: How certain should one be that the mean tree accuracy is higher for PC-chord grammars than for TPC-chord grammars? The mean tree-accuracy difference between two models, denoted by Δ, is therefore considered a random variable and the Bayesian bootstrap is applied to estimate its distribution.

Figure 9.5 shows the distributions of mean tree-accuracy differences for the single-component grammars and the random baseline. The black line at 0 marks the point where the models are not distinguishable by tree accuracy. In this and the following figures, the difference is calculated by subtraction of the mean tree accuracy of the model named at the left side from the column model named at the top.

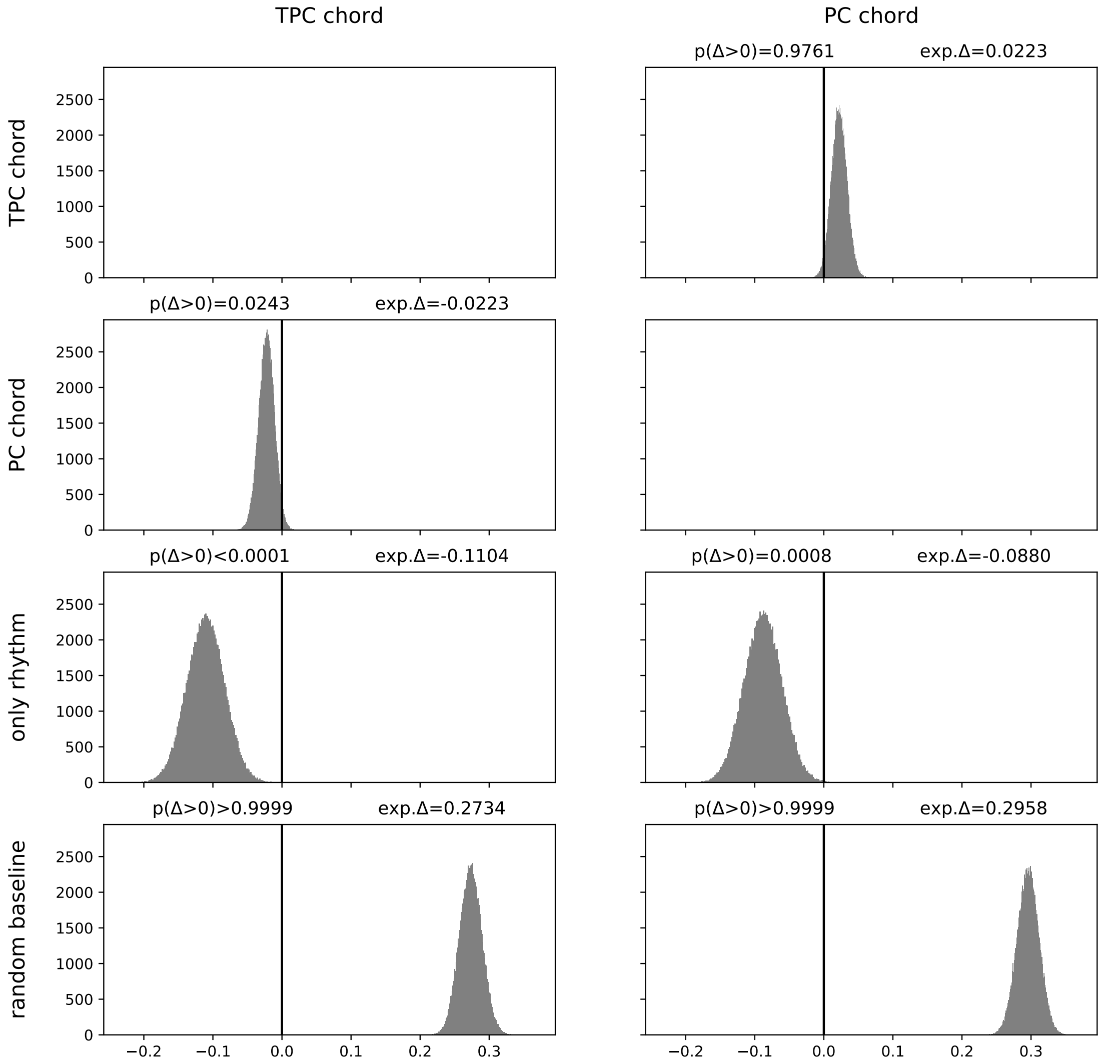


Figure 9.5 – Mean tree-accuracy difference (Δ) estimation for single-component models of harmony and rhythm (100,000 bootstrap samples each). The x-axes show the accuracy differences and the y-axes show the number of re-sampled datasets. All plots are barplots with 150 bins. Each plot shows how much higher the mean tree accuracy of the column model (indicated at the top) is.

The plot in the first row shows the improvement in tree accuracy of the PC-chord grammar in comparison to the TPC-chord grammar — the improvement caused by transpositional invariance. The expected improvement is with 2.2% not very high, but since the variance of the distribution is small, the probability that the PC-chord grammar yields a higher mean tree accuracy than the TPC-grammar is greater than 97%. In odds this means that the PC-grammar is over 40 times more likely to yield higher tree accuracies than the TPC-grammar. This is strong evidence for the PC-chord grammar model (Kass and Raftery, 1995). The plots in the third and the fourth row show the very strong evidence for the facts that both PC-chord and TPC-chord grammar models yield lower tree accuracies than the single-component model for rhythm and higher accuracies than the random baseline.

Analogously to Figure 9.5, Figure 9.6 shows the distributions of mean tree-accuracy differences between product grammars and between product grammars and their components. The following first focuses on the plot in the first row. With the joint rhythm model, the expected improvement in tree accuracy caused by the transpositional-invariant parameterization is with 1.4% lower than the expected improvement without jointly modeling rhythm. The evidence for a positive effect of transpositional invariance is weaker than for the single-component grammars of harmony, but the PC-chord product grammar is still more than 15 times more likely to yield higher tree accuracies than the TPC-product grammar.

The accuracy measures compare the tree predictions of the models to the treebank trees. Bootstrapped Bayes factors are additionally reported to compare the PC-chord and the TPC-chord grammar models with and without jointly modeling rhythm. Those Bayes factors quantify how much more likely the PC-chord grammar is than the TPC-chord grammar, with respect to the chord sequences (not the trees) of the treebank tunes. Figure 9.7 shows the common logarithms of the Bayes factors. Since the probability that both Bayes factors are greater than 10^{400} is nearly equal to 1, the results provide very strong evidence in favor of the transpositionally invariant PC-chord grammar.

9.4.2 Jointly modeling rhythm

The evidence for the first hypothesis, that jointly modeling rhythm improves grammar models of harmony, is discussed in the following. Since MLP and Bayes factors cannot be used to compare grammar models with different terminals, only tree accuracy is used to compare the PC-chord and TPC-chord product grammars to their harmony components. The results are again analogous for dependency accuracy.

Figure 9.6 shows mean tree-accuracy differences to compare the PC-chord and TPC-chord product grammars to their components and to the random baseline. For both PC-chord and TPC-chord nonterminals, the evidence that the corresponding product grammar yields higher tree accuracies than both of its component grammars is very strong. The expected improvement in tree accuracy by jointly modeling rhythm is larger than 15%. Notably, the component grammar for rhythm outperforms the component grammar for harmony as shown

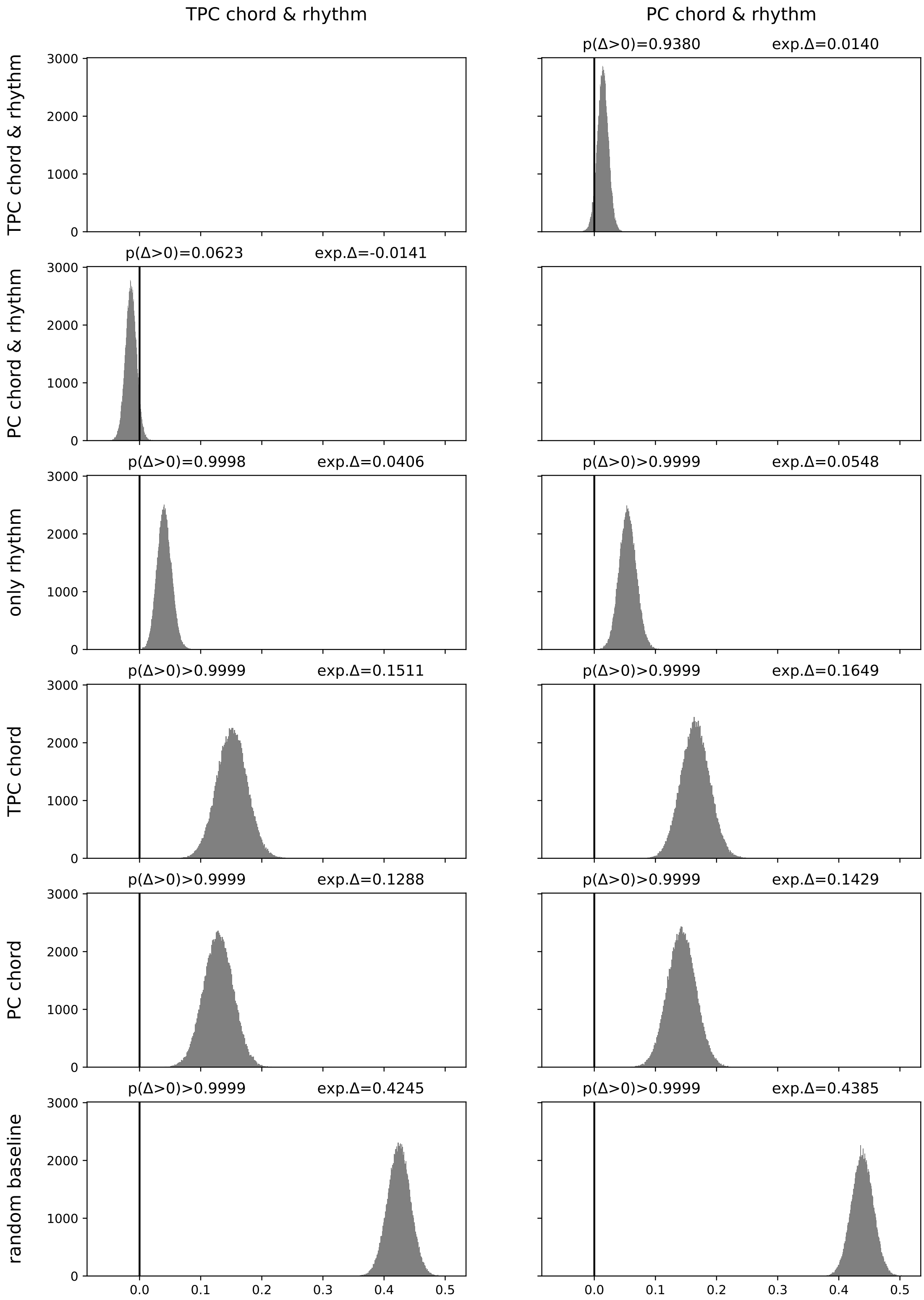


Figure 9.6 – Mean tree-accuracy difference (Δ) estimation for joint models of harmony and rhythm. See the caption of Figure 9.5 and the main text for more details.

in Figure 9.5. The product grammars, however, outperform the single-component rhythm grammar by about 4%–5%.

Why does jointly modeling rhythm improve the tree predictions so much? As described in Section 5.1, product grammars can exploit the regular rhythm of harmonic phrases in Jazz standards. Since the rhythm grammar favors simple split-ratios of chord durations, its predictions are balanced trees. For instance, Figure 9.14 shows the tree prediction of the rhythm grammar for the Jazz standard *Summertime*. All branches of this tree split chord durations with the same split ratio of 1/2.

As a result, product grammars also favor balanced trees. Since the lengths of the treebank sequences are constant, the balancedness of a model's tree predictions can be quantified by the average tree height of a prediction. Table 9.1 shows the average tree heights for all models and baselines. The average tree height of single-component grammars for harmony is with about 12.5 almost twice as large than for the corresponding product grammars with an average tree height of about 7. In contrast, the single-component grammar for rhythm yields slightly smaller trees than the product grammars. Its average tree height is 6.5. Since the average tree height of the treebank is about 7.5, the heights of the tree predictions of the product grammars are closed to the heights of the treebank trees. The impact of balancedness on phrase prediction is qualitatively discussed in the next section.

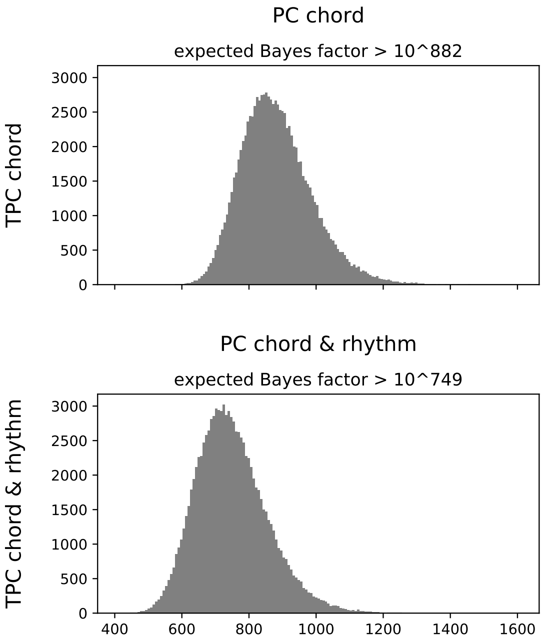


Figure 9.7 – Logarithms of Bayes factors in favor of PC-chord single-component and product grammars, respectively. See the caption of Figure 9.5 and the main text for more details. Note that the plot at the bottom shows only a rough approximation of the Bayes factors because rewrite probabilities of product rules are not represented exactly.

9.5 Qualitative analysis of tree predictions

9.5.1 Model comparison using the example of *Summertime*

Figures 9.9–9.14 show the treebank tree and the tree predictions of all models for the Jazz standard *Summertime.* A lead sheet of this tune is shown in Figure 9.8; the chords are from the iRealPro dataset and the melody is based on the version of the *Real Bool Volume II* (no credits or date existent). *Summertime* was chosen to illustrate the models and their predictions, because it is a short tune that has compound form. The formal structure of the tune is that of a *period*; it is a common structure in Jazz standards and, for example, also in Western classical music of the common practice period (Caplin, 1998). Furthermore, the harmonic complexity of *Summertime* is at a medium level. It is not too complex, but includes applied dominants such as $A^7$and a tonicization to the relative key C major. All tree predictions of all models can be accessed online.[4]

The treebank analysis is considered briefly before the tree predictions are discussed. Figure 9.9 shows the expert analysis of the *Summertime.* The syntax tree structures the tune into four phrases which correspond to the subtrees whose roots are generated at the second tree level (counting top-down where the root of the tree is on level 0). Each phrase corresponds to one form part. The formal structure of the tune is an ABAC structure in which AB forms an antecedent and AC a consequent phrase of a period. Since the antecedent phrase ends in a half cadence, it is denoted by an open constituent. The harmonic structure of the A part is simple. The harmony changes from the tonic $Am^7$to the dominant $E^7$and back to the tonic. The A part thus constitutes a tonic phrase. In contrast, the B part is a dominant phrase. It starts with an applied dominant to the chord on the fourth scale degree Dm^7. Since $Dm^7$is on a hypermetrically strong position, the rhythmic structure of its applied dominant $A^7$is that of a harmonic upbeat. After Dm^7, the phrase proceeds via more applied dominants to the fifth scale degree — the chord E^7. Then, the tune restarts with a repetition of the A part and closes with the C part. The C part starts with a tonicization of the relative key C major. The rhythm of the chords $D^7$and $G^7$is thereby that of a harmonic upbeat, analogous to the first chord $A^7$of the B part. After the $C^\triangle$chord, the harmony changes back to the key of A minor and the tune closes with a II-V-I progression. Note that this is the only occurrence of a II-V-I progression in the tonic key.

The tree prediction of the single-component TPC-chord grammar is shown in Figure 9.10. Since this grammar does not model rhythm, the tree is unbalanced and appears to be chaotic. The grammar correctly identifies the I II V I progression at the end of the tune, but overly emphasizes the importance of the chords $D^7$and G^7. In fact, it analyses the tune to be in C major despite the many occurrences of the chord Am^7. The reason for this could be that 11 tunes analysed in the treebank have $C^\triangle$as a tonic chord and only 2 tunes have Am^7as tonic.[5] From the 4 phrases which correspond to the formal parts A, B, A, and C, the TPC-chord

[4] https://github.com/dharasim/LearnabilityJazzGrammar

[5] These low numbers result from the fact the treebank tunes in C major commonly end with a C^6chord. Also, the

grammar only analyses the B part as a constituent. As a consequence, the high-level structure of the prediction is badly shaped.

The tree prediction of the single-component PC-chord grammar is shown in Figure 9.11. This tree is already less chaotic than the TPC-chord grammar's prediction. The PC-chord grammar recognizes the final I II V I progression and identifies the first A phrase as a constituent. However, it does not recognize the remaining phrases and also analyses the tonic of the tune as $C^{\triangle}$.

The tree predictions of the TPC-chord and the PC-chord grammar with the joint rhythm model are shown in Figures 9.12 and 9.13. Since these trees are very similar, they are discussed in direct comparison. First of all, both grammars recognize the tune's key and its high-level structure consisting of two phrases. This is enabled by the joint rhythm model, because those phrases are of equal length and the rhythm model favors simple splits of chord durations. The analysis of the first half of the tune as a constituent implies that the product grammars are able to successfully learn and apply the concept of open constituents. In fact, the product grammars overgeneralize this concept. For instance the first three chords are predicted to form an open constituent which does not correspond to the structure of *Summertime*; the melody of the tune clearly indicates that the first E^7chord resolves immediately into the following Am^7. The usage and overgeneralization of open constituents in *Summertime* is characteristic for the product grammar models; it happens similarly in the prediction of other tunes such as *All of me*, *Struttin' with some Barbeque*, and *A beautiful friendship*.

One notable difference between the two tree predictions is the analysis of the A^7chord. The PC-chord product grammar correctly identifies it as an applied dominant of Dm^7while the TPC-chord product grammar relates it to the tonic chord Am^7. This mistake of the TPC-chord model results from the fact that the grammars do not explicitly represent which chords are tonic chords — while a prolongation of a minor chord with a dominant-seventh chord would be very uncommon for tonics, it is more plausible for second scale degrees which change from a minor to a double-dominant chord.

The tree prediction of the single-component rhythm grammar is shown in Figure 9.14. Since it only uses chord duration split ratios of 1/2, it shows rather the metrical and hypermetrical structure of the tune instead of its harmonic rhythm. The shape of the rhythm grammar's tree prediction is very similar to the prediction of the TPC-chord product grammar. This suggests that the PC-chord product grammar is not confident enough to overrule the rhythm grammar. In contrast, the PC-chord grammar occasionally overrules the rhythm grammar, for example to identify A^7as an applied dominant or to correctly identify the second A part of the tune as a constituent.

key of C minor if for example more common as A minor.

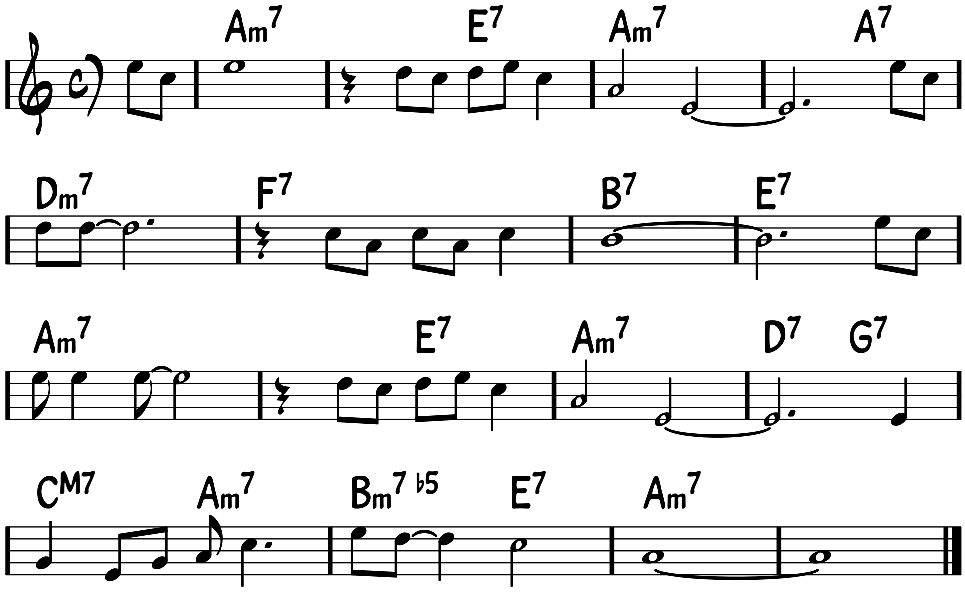


Figure 9.8 – Lead-sheet transcription of the Jazz standard *Summertime* in the key of A minor. The transcription was created using LilyPond (http://lilypond.org/) with LilyJAZZ fonts (https://github.com/OpenLilyPondFonts/lilyjazz). The chords are from the iRealPro dataset and the melody is based on the version of the *Real Bool Volume II*.

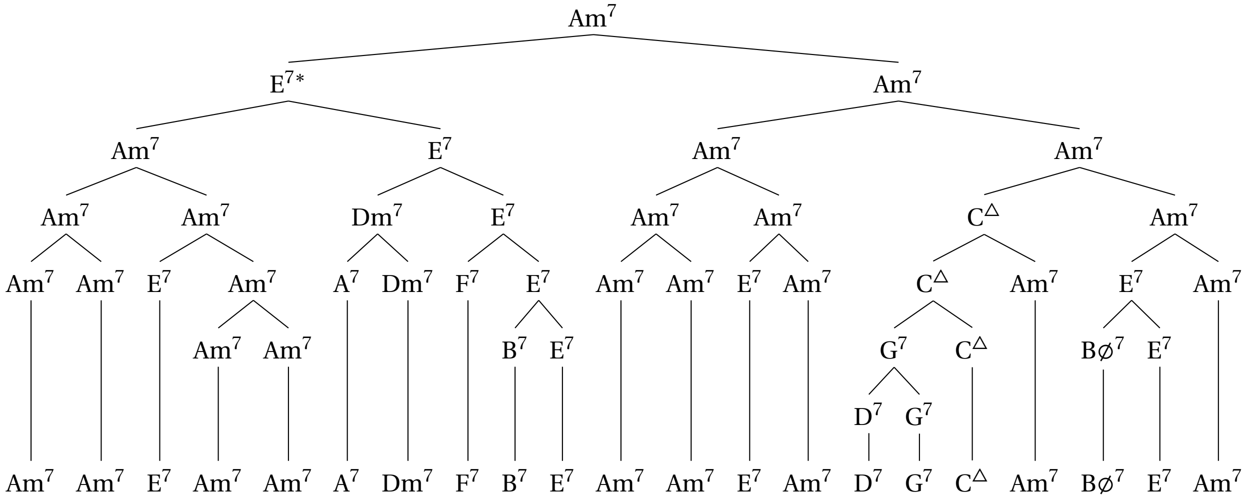

Am^7
E^{7*} Am^7
Am^7 E^7 Am^7 Am^7
Am^7 Am^7 Dm^7 E^7 Am^7 Am^7 C$^\triangle$ Am^7
Am^7 Am^7 E^7 Am^7 A^7 Dm^7 F^7 E^7 Am^7 Am^7 E^7 Am^7 C$^\triangle$ Am^7 E^7 Am^7
Am^7 Am^7 B^7 E^7 G^7 C$^\triangle$ Bø^7 E^7
D^7 G^7
Am^7 Am^7 E^7 Am^7 Am^7 A^7 Dm^7 F^7 B^7 E^7 Am^7 Am^7 E^7 Am^7 D^7 G^7 C$^\triangle$ Am^7 Bø^7 E^7 Am^7

Figure 9.9 – Treebank tree analysis of the tune *Summertime*. The asterisk which marks the left subtree as an open constituent is part of the treebank analysis, but not used during learning.

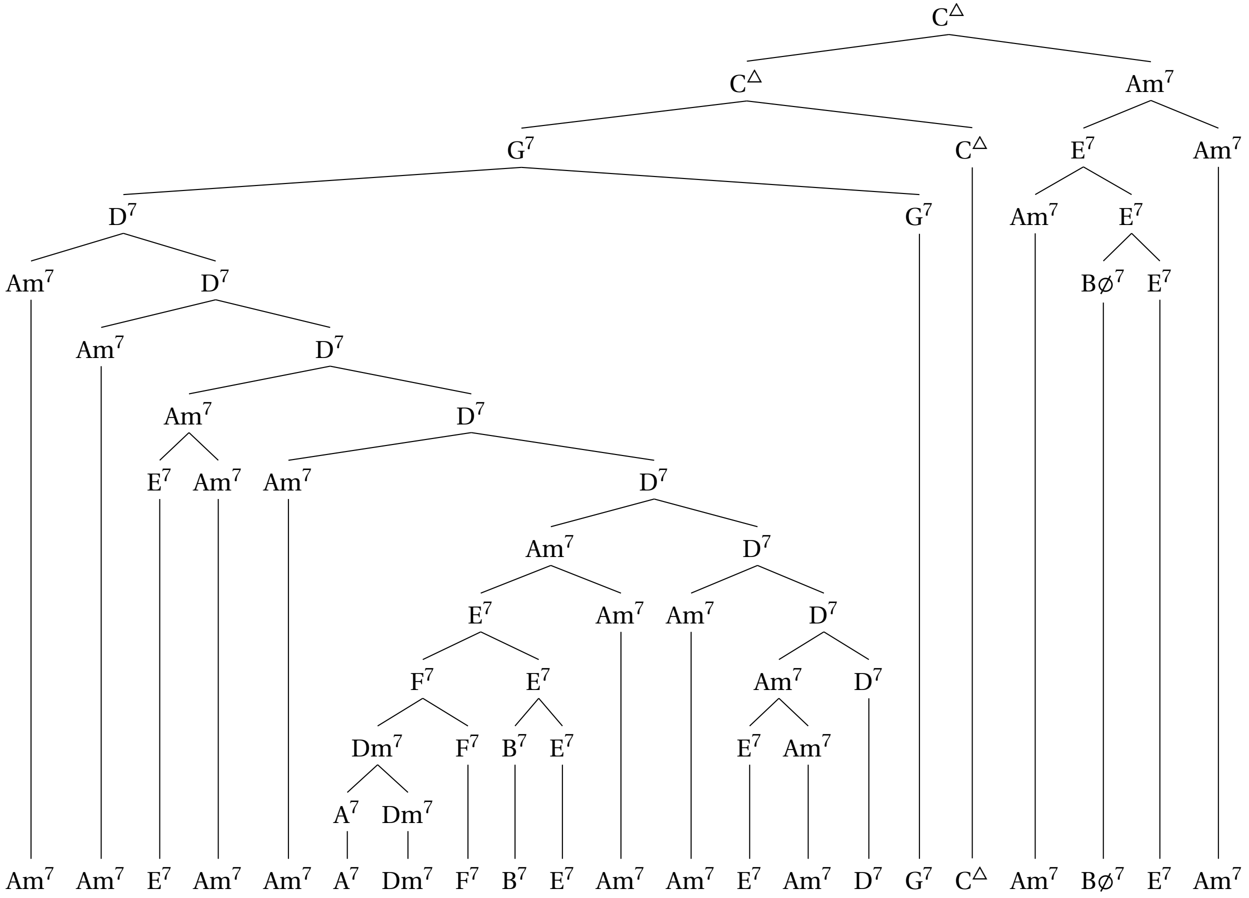


Figure 9.10 – Tree prediction of the TPC-chord grammar (single-component, supervised) for the tune *Summertime*.

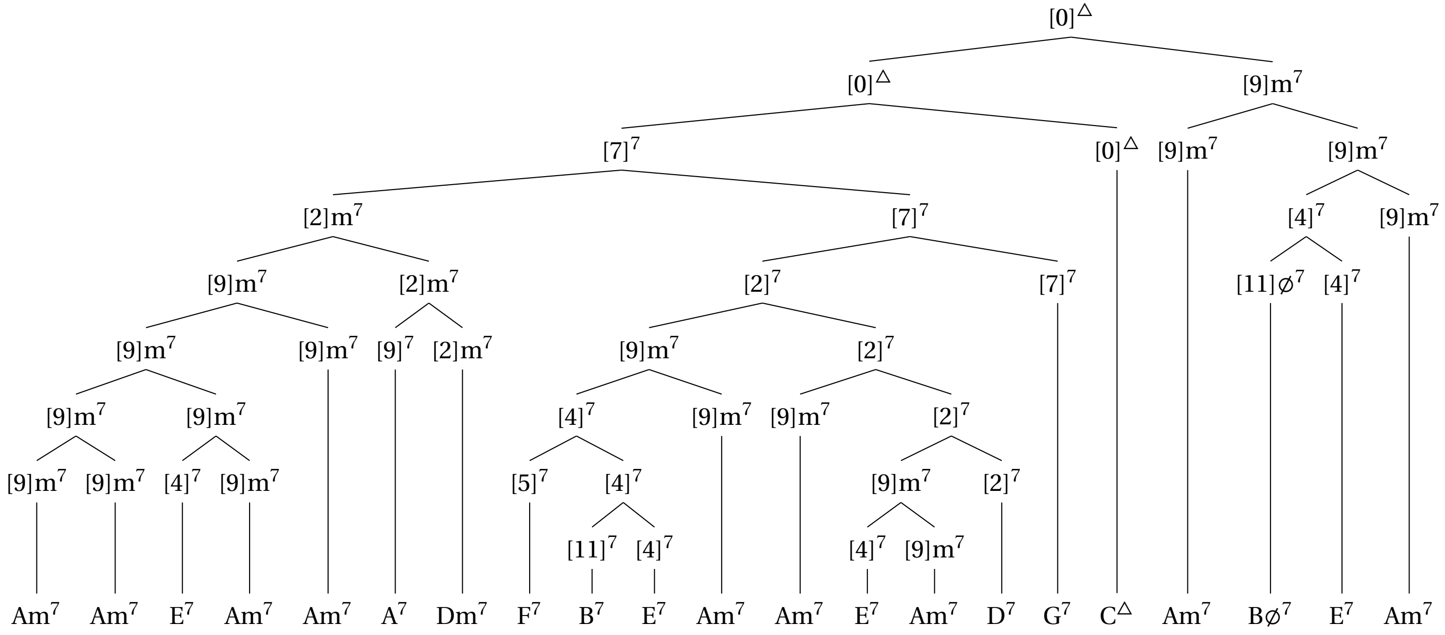


Figure 9.11 – Tree prediction of the PC-chord grammar (single-component, supervised) for the tune *Summertime*.

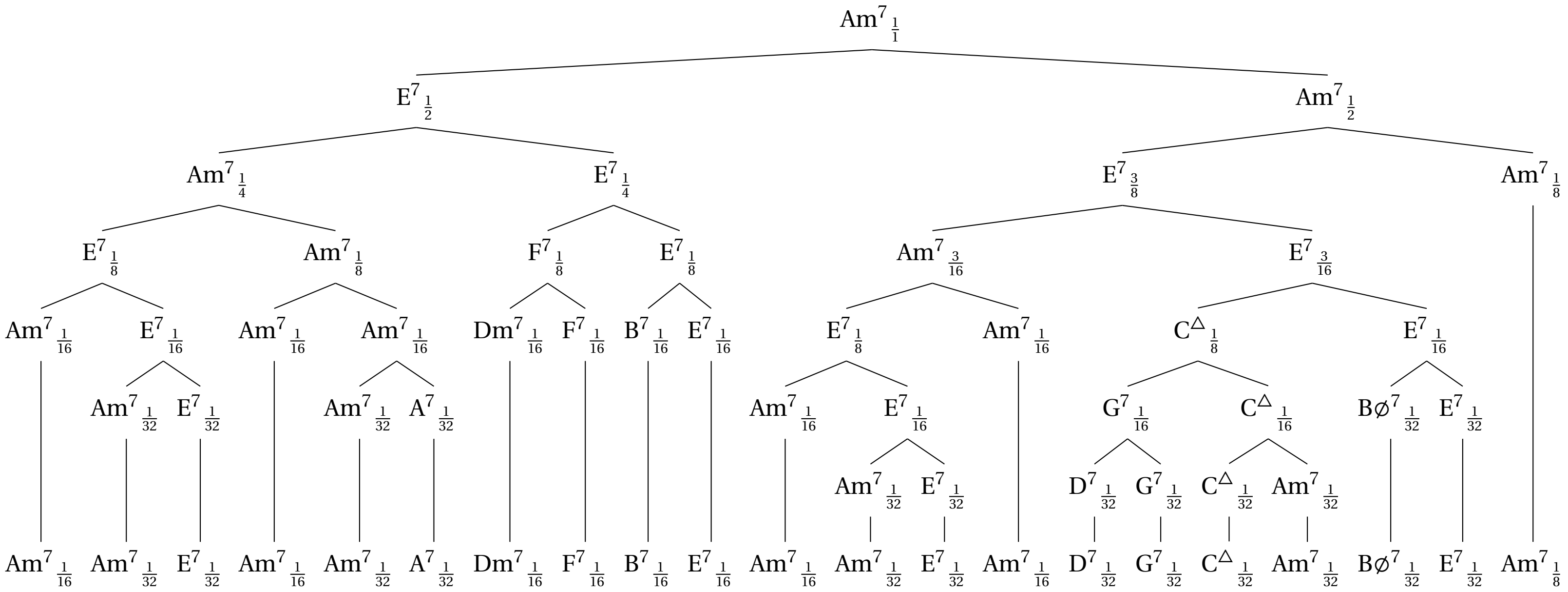


Figure 9.12 – Tree prediction of the TPC-chord product grammar (jointly modeling rhythm, supervised) for the tune *Summertime*.

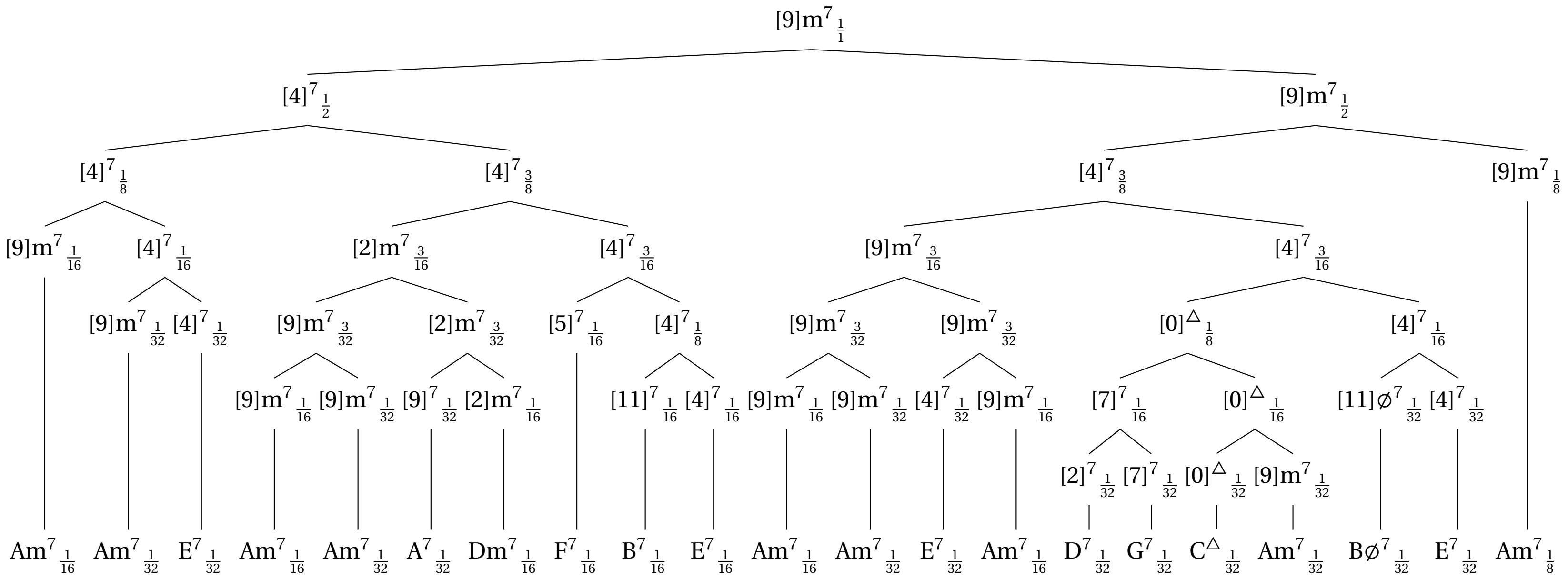

$[9]m^7_{\frac{1}{1}}$
$[4]^7_{\frac{1}{2}}$ $[9]m^7_{\frac{1}{2}}$
$[4]^7_{\frac{1}{8}}$ $[4]^7_{\frac{3}{8}}$ $[4]^7_{\frac{3}{8}}$ $[9]m^7_{\frac{1}{8}}$
$[9]m^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{16}}$ $[2]m^7_{\frac{3}{16}}$ $[4]^7_{\frac{3}{16}}$ $[9]m^7_{\frac{3}{16}}$ $[4]^7_{\frac{3}{16}}$
$[9]m^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$ $[9]m^7_{\frac{3}{32}}$ $[2]m^7_{\frac{3}{32}}$ $[5]^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{8}}$ $[9]m^7_{\frac{3}{32}}$ $[9]m^7_{\frac{3}{32}}$ $[0]^{\triangle}_{\frac{1}{8}}$ $[4]^7_{\frac{1}{16}}$
$[9]m^7_{\frac{1}{16}}$ $[9]m^7_{\frac{1}{32}}$ $[9]^7_{\frac{1}{32}}$ $[2]m^7_{\frac{1}{16}}$ $[11]^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{16}}$ $[9]m^7_{\frac{1}{16}}$ $[9]m^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$ $[9]m^7_{\frac{1}{16}}$ $[7]^7_{\frac{1}{16}}$ $[0]^{\triangle}_{\frac{1}{16}}$ $[11]\varnothing^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$
$[2]^7_{\frac{1}{32}}$ $[7]^7_{\frac{1}{32}}$ $[0]^{\triangle}_{\frac{1}{32}}$ $[9]m^7_{\frac{1}{32}}$
$Am^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{32}}$ $E^7_{\frac{1}{32}}$ $Am^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{32}}$ $A^7_{\frac{1}{32}}$ $Dm^7_{\frac{1}{16}}$ $F^7_{\frac{1}{16}}$ $B^7_{\frac{1}{16}}$ $E^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{32}}$ $E^7_{\frac{1}{32}}$ $Am^7_{\frac{1}{16}}$ $D^7_{\frac{1}{32}}$ $G^7_{\frac{1}{32}}$ $C^{\triangle}_{\frac{1}{32}}$ $Am^7_{\frac{1}{32}}$ $B\varnothing^7_{\frac{1}{32}}$ $E^7_{\frac{1}{32}}$ $Am^7_{\frac{1}{8}}$

Figure 9.13 – Tree prediction of the PC-chord product grammar (jointly modeling rhythm, supervised) for the tune *Summertime*.

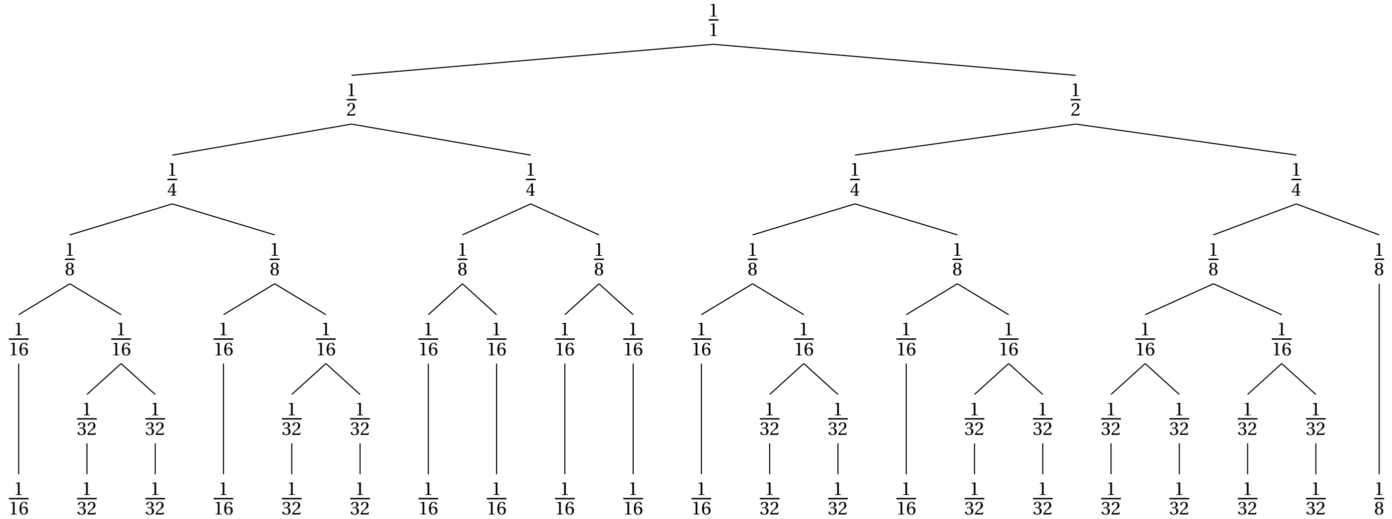


Figure 9.14 – Tree prediction of the rhythm grammar (single-component, supervised) for the tune *Summertime*.

9.5.2 Prediction of initial tonics and turnarounds

Since none of the applied grammar models has explicit representations for keys or tonic chords, some tonic prolongations are confused as preparations. This is particularly common for initial tonic chords which prolong tonics far apart. Consider for example the Jazz standard *Broadway* in the key of B♭ major. It is a tune with an AABA structure whose A part follows the chord progression $\mathrm{B}\flat^{\triangle}$ $\mathrm{E}\flat^{7}$ Cm^{7} F^{7} $\mathrm{B}\flat^{\triangle}$. Figure 9.15 shows the full tree prediction of the TPC-chord product grammar (the PC-chord product grammar's prediction is identical, but harder to read). Because of the the fifth relationship between the roots of the two initial chords, the product grammars analyse $\mathrm{B}\flat^{\triangle}$ as a preparation of $\mathrm{E}\flat^{7}$. They fail to identify the first chord as a tonic. Interestingly, the single-component grammars for harmony correctly identify the initial chord as a tonic, but are unable to identify the formal (deep-level) structure of the tune. Similar misinterpretations happen for example in *My melancholy baby*, *The good life*, and *Take the A train*.

A second mistake common in product grammar predictions is the wrong attachment of turnarounds. For instance in the tree analysis of the tune *Broadway* shown in Figure 9.15, the turnaround $\mathrm{G}^{7}\mathrm{Cm}^{7}\mathrm{F}^{7}$which leads to the second A part is not attached to the second but to the first A part. The first A part is consequently analyzed as an open constituent which is, however, neither supported by the harmonic rhythm nor the melody of the tune. In fact, the doubling of the harmonic rhythm after the arrival on the tonic in measure seven of an 8-measure phrase is a strong indicator of a turnaround. The melody plays the tonic note over the tonic chord and the chords of the turnaround. This is also evidence against an open constituent for which a melody note on the second scale degree is for example more common. However, the melodic information is not accessible by the grammar models. Because of the harmonic rhythm and the melody, the turnaround constitutes a harmonic upbeat to the second A part. Similar misattachments happen for example in *I love Paris*, *Remember*, and *Take the "A" train*.

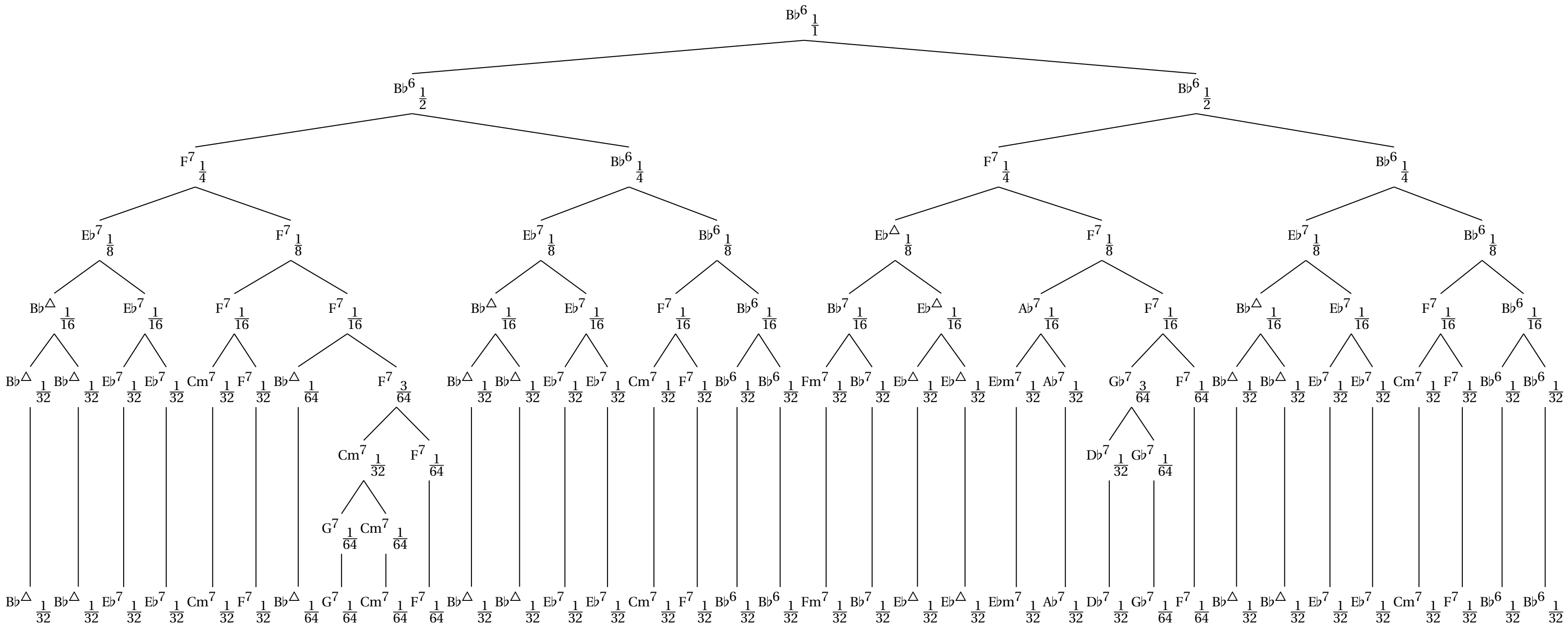


Figure 9.15 – Tree prediction of the TPC-chord product grammar (jointly modeling rhythm, supervised) for the tune *Broadway*. A larger version of this tree is available online at https://github.com/dharasim/LearnabilityJazzGrammar/blob/master/supervised-learning/tpc-supervised-product.pdf on page 112.

9.5.3 Tunes in ternary form

The single-component grammar for rhythm is already able to correctly detect ternary forms. This is surprising, because chord duration splits of 1 : 1 (1 to 1) are far more common than splits of 2 : 1. Still, the rhythm grammar correctly predicts the 3 parts of tunes in ternary form without consideration of the chord symbols. This is for example the case in the Jazz standard *Song for my father*. Its treebank analysis is shown in Figure 9.16, and the prediction of the rhythm grammar is shown in Figure 9.17. The tune is presented in the key of F minor; its formal structure is AAB.

The reason why the rhythm grammar analyzes *Song for my father* as having a ternary instead of a binary form (in contrast to most tunes analyzed in the treebank that have binary forms) is surprisingly simple. Since 3 is a divisor of 24, the number of the tune's measures, and each chord is either a half, one, or two measures long, the rhythm grammar must split the constituent durations 1 : 2 or 2 : 1 at least once. Since 1 : 1 duration splits are most common, the tree that uses a spit 1 : 2 or 2 : 1 only once is the most probably analysis. A analysis which structures the tune into a binary structure has lower probability, because then a duration split of 1 : 2 or 2 : 1 would have to be applied twice — in both parts of the binary form.

Other Jazz standards in the treebank which have a ternary form are for instance *Why don't you do right?*, *Mr. P.C.*, and *Footprints*.

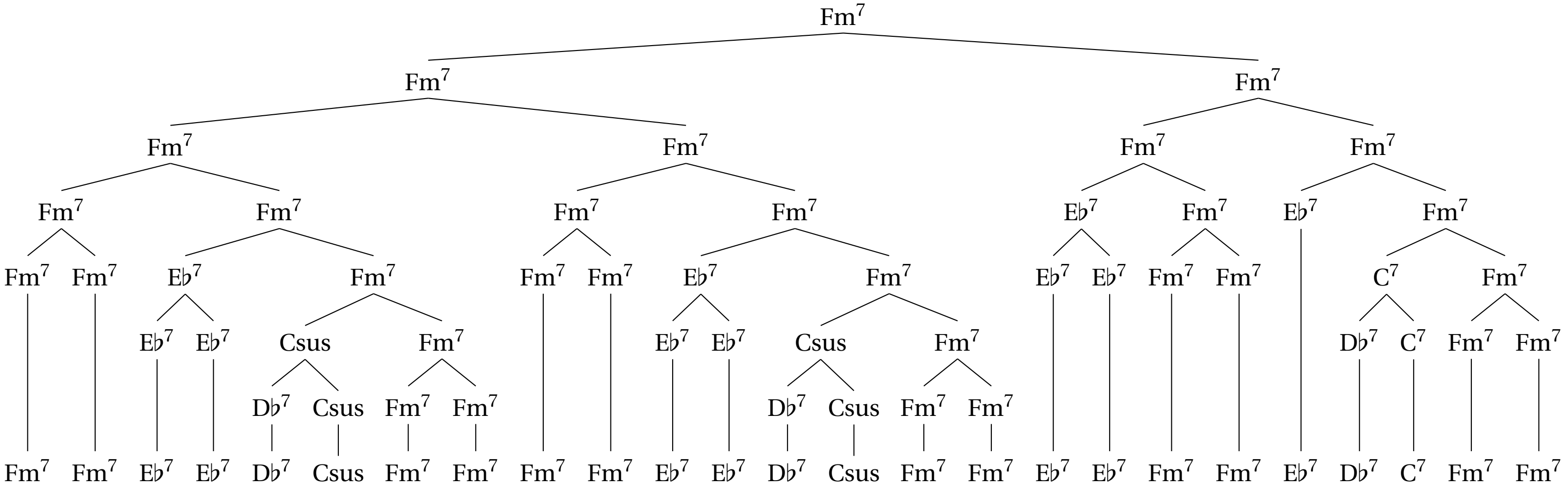

Fm7
Fm7 Fm7
Fm7 Fm7 Fm7 Fm7
Fm7 Fm7 Fm7 Fm7 E♭7 Fm7 E♭7 Fm7
Fm7 Fm7 E♭7 Fm7 Fm7 Fm7 E♭7 Fm7 E♭7 E♭7 Fm7 Fm7 C7 Fm7
E♭7 E♭7 Csus Fm7 E♭7 E♭7 Csus Fm7 D♭7 C7 Fm7 Fm7
D♭7 Csus Fm7 Fm7 D♭7 Csus Fm7 Fm7
Fm7 Fm7 E♭7 E♭7 D♭7 Csus Fm7 Fm7 Fm7 Fm7 E♭7 E♭7 D♭7 Csus Fm7 Fm7 E♭7 E♭7 Fm7 Fm7 E♭7 D♭7 C7 Fm7 Fm7

Figure 9.16 – Treebank tree analysis of the tune *Song for my father*.

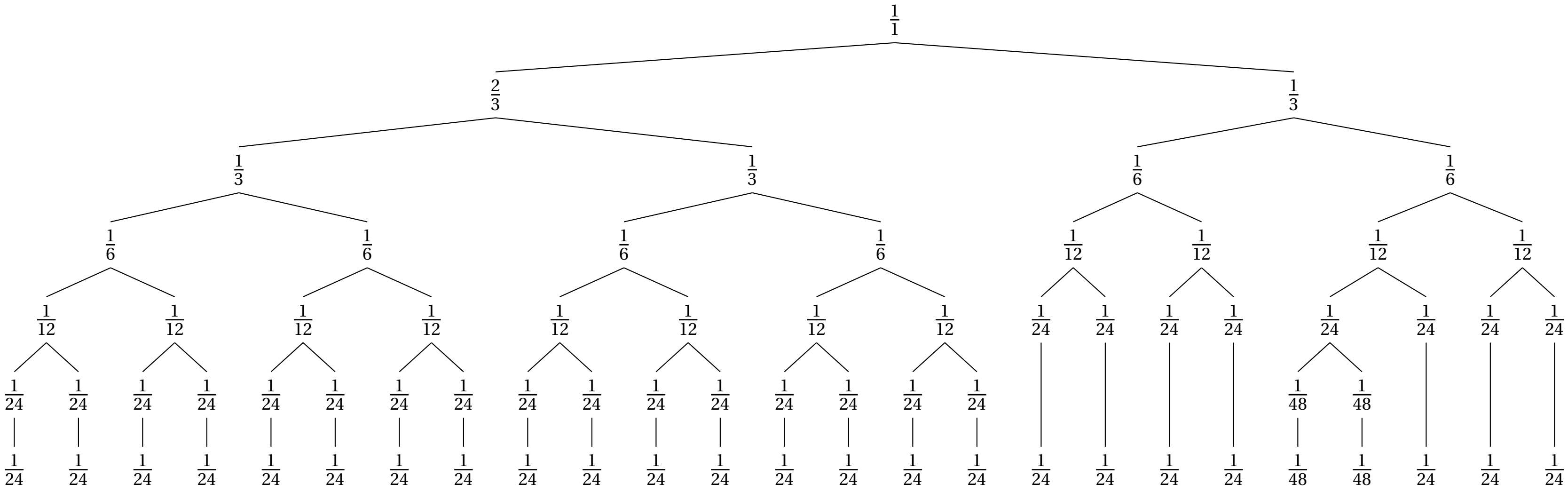


Figure 9.17 – Tree prediction of the rhythm grammar (single-component, supervised) for the tune *Song for my father*.

10 Unsupervised grammar learning

The central research question studied in the unsupervised grammar-learning experiments presented in this chapter is:

> What prior knowledge enables an ideal learner to induce a harmony grammar of high quality from the observation of chord sequences?

By using computational learning simulations, one can only study what prior knowledge is sufficient for learning a harmony grammar. That is, an upper bound of the required prior knowledge is determined. The goal of the experiments is to find minimal prior knowledge that is cognitively plausible in order to increase the strength of the learnability argument. Before the execution of the experiments, it was hypothesized that it is possible to learn a grammar of similar quality to the grammar obtained in the supervised experiments presented in Chapter 9 by using the domain-general assumption of headedness and by jointly modeling harmony and rhythm. Furthermore, the assumption of goal-directedness, which reduces the size of the space of possible grammars, was hypothesized to ease learnability and preserve quality.

The task of inferring a grammar, including all symbols, rules, and probabilities, from the mere observation of sequential data is known as *grammar induction*. Computational grammar induction is a challenging task that originated from the study of natural language; it has been considered in computational linguistics for over 30 years where it is still an active research topic (e.g., Kim et al., 2019; Golland et al., 2012; Johnson et al., 2007b; Klein and Manning, 2002; Lari and Young, 1990). Grammar induction is also studied in formal language theory (e.g., Clark, 2013a; Yoshinaka, 2011; Cohn et al., 2010; López et al., 2004). For music there are, however, only few studies known that consider grammar induction. Tsushima et al. (2017) induce a PCFG with up to 20 induced nonterminal symbols for chord-symbol harmonization of melodies. Their results are a proof of concept, but the quality of the induced grammar is much worse than those of grammars learned from expert tree analyses. Tsushima et al. (2018) performed grammar induction without modeling rhythm, similar to Tsushima et al. (2017), and showed that grammar induction can benefit from an initialization using a hidden-Markov model. Their results crucially rely on a reduction of the number of terminals to 10, 20, and

50 by grouping rare terminals into one artificial terminal. In contrast, this study induces grammars without a pre-grouping of terminals or hidden-Markov initialization by jointly modeling harmony and rhythm and constraining the form of rules to duplication and headed rules. Déguernel et al. (2019) use a PCFG to model musical form on multiple levels — such as chord transitions, harmonic phrases, and formal sections — with the goal of automatic improvisation. They learn a grammar by iteratively grouping correlated nonterminals and evaluate their results qualitatively with professional musicians. In contrast, this study applies variational Bayesian inference to approximate a distribution over probabilistic grammars of harmonic syntax.

Two experiments are presented in this chapter, Experiment A and Experiment B. In Experiment A, multiple models are tested for their grammar induction capabilities. In addition to the grammar models used for supervised learning in Chapter 9, we test alternative versions that can make use of prior knowledge such as the goal-directedness of functional harmony. Such prior knowledge guides the learning agent by reducing the space of possible grammars. The results of Experiment A show that the goal-directed, transpositionally invariant grammar that jointly models rhythm performs best. However, the rhythm grammar heavily overuses the chord-duration-split ratio 1/2 and constantly overrules the component grammar for harmony — the rhythm grammar is very confident that it is right but it actually is wrong, stuck in a bad local optimum.

The goal of Experiment B is to balance the grammar components and to find a cognitively plausible prior distribution that regularizes the grammar component for rhythm. Such a prior distribution is successfully found: with additional prior preference for simple duration-split ratios, harmonic grammars of good quality are learnable from the observation of chord sequences. Only goal-directed grammars that jointly model rhythm were considered in Experiment B, because they performed best in Experiment A.

10.1 Inference and evaluation

For both experiments and each grammar model, learning is performed by starting with a large set of equally probable rules representing all structural possibilities for that model. This initialization represents the space of grammars the learner is able to acquire. The rewrite-rule probabilities are then learned using variational Bayesian inference. After learning, the set of rules can be thought of as partitioned into a subset of actual grammar rules (rules with relatively high probability) and rules with very low probability whose function it is to make the grammar robust against sequences of uncommon structure. This reduces the problem of learning the rules of a grammar to a parameter-estimation problem. The partition of the rules is, however, not explicitly considered in this study.

The unsupervised learning of the rewrite probabilities is described in detail in Chapter 7 and summarized in Section 7.4. The 150 trees of the Jazz harmony treebank (*test set* in the following) were used to evaluate the trained grammar models. For learning, each model observed 300 Jazz chord sequences of tunes contained in the iRealPro dataset but not analyzed in the Jazz harmony treebank (*training set* in the following). Using the notation introduced in the previous chapters, the probability of a grammar's rewrite probabilities $\boldsymbol{\theta}$ conditioned on the terminal sequences $\bar{\boldsymbol{w}}$ of the training set is given by marginalization of the derivation trees $\bar{\boldsymbol{r}}$,

$$p(\boldsymbol{\theta} \mid \bar{\boldsymbol{w}}) = \sum_{\bar{\boldsymbol{r}} \in \mathscr{R}^I} p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}}), \tag{10.1}$$

where I denotes the number of observed sequences. Since an exact computation of the probability $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ is not tractable, the distribution $p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ is approximated by a distribution $q(\boldsymbol{\theta}, \bar{\boldsymbol{r}})$ from a simpler family which assumes independence of rewrite probabilities $\boldsymbol{\theta}$ and derivations $\bar{\boldsymbol{r}}$,

$$p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}}) \approx q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) = q(\boldsymbol{\theta})\ q(\bar{\boldsymbol{r}}). \tag{10.2}$$

The optimal parameters of the approximating distributions $q(\boldsymbol{\theta})$ and $q(\bar{\boldsymbol{r}})$ are then obtained iteratively by minimizing the KL-divergence

$$\mathrm{KL}\big(q(\boldsymbol{\theta}, \bar{\boldsymbol{r}}) \,\big\|\, p(\boldsymbol{\theta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})\big) \tag{10.3}$$

using coordinate ascent variational inference (CAVI).

A generalization of CAVI to stochastic variational inference was adopted that uses batch updates instead of the whole training set in the iterative optimization steps (Hoffman et al., 2013). The batch size was chosen as 30. Since CAVI only converges to a local optimum, the parameter inference was performed 10 times for each model and the parameter setting was chosen that yielded the maximal predictive probability of the training set. Since CAVI is much more efficient than more general optimization algorithms based on stochastic gradient descent, convergence was always obtained within the first five epochs as prior experiments showed. In fact, most of the learning happens already in the first epoch (i.e., during the first 10 batches). Each trial was therefore run for 5 epochs. Similarly to the supervised experiments, the learning of product grammars was performed by propagating the expected rule counts to the rule distributions of the component grammars.

The tree prediction of the grammar models and the quantitative evaluation is analogous to the supervised experiments presented in Chapter 9. For a chord sequence $\boldsymbol{v}$ from the test set, each grammar model predicts a derivation tree $\boldsymbol{r} \in \textsc{Der}(\boldsymbol{v})$ by maximizing the posterior probability $p(\boldsymbol{r} \mid \bar{\boldsymbol{w}})$,

$$\operatorname*{arg\,max}_{\boldsymbol{r} \in \textsc{Der}(\boldsymbol{v})} p(\boldsymbol{r} \mid \bar{\boldsymbol{w}}) = \operatorname*{arg\,max}_{\boldsymbol{r} \in \textsc{Der}(\boldsymbol{v})} \int_{\boldsymbol{\theta}} p(\boldsymbol{r} \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta} \mid \bar{\boldsymbol{w}}) \approx \operatorname*{arg\,max}_{\boldsymbol{r} \in \textsc{Der}(\boldsymbol{v})} \int_{\boldsymbol{\theta}} p(\boldsymbol{r} \mid \boldsymbol{\theta})\ q(\boldsymbol{\theta}). \tag{10.4}$$

The Mean Log Predictive (MLP) of the terminal sequences $\bar{\boldsymbol{v}}$ of the test set is given by

$$\mathrm{MLP}(\bar{\boldsymbol{v}}) = \frac{1}{I} \sum_{i=1}^{I} \log p(\boldsymbol{v}^i \mid \bar{\boldsymbol{w}}) \tag{10.5}$$

where $I = 150$ is the size of the test set (the number of tree-annotated chord sequences) and

$$p(\boldsymbol{v}^i \mid \bar{\boldsymbol{w}}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{v}^i \mid \boldsymbol{\theta})\ p(\boldsymbol{\theta} \mid \bar{\boldsymbol{w}}) \approx \int_{\boldsymbol{\theta}} p(\boldsymbol{v}^i \mid \boldsymbol{\theta})\ q(\boldsymbol{\theta}) = \int_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) \sum_{\boldsymbol{r}^i \in \mathrm{DER}(\boldsymbol{v}^i)} p(\boldsymbol{r}^i \mid \boldsymbol{\theta}). \tag{10.6}$$

10.2 Experiment A: model comparison for grammar induction

Experiment A investigates the grammar-inference capabilities of multiple grammar models. The models tested in the experiments vary along three dimensions: the choice of the harmony grammar's nonterminals, the form of allowed rules, and whether harmony is modeled jointly with rhythm or not.

Three types of nonterminals are tested for the harmony component of the grammar models: Tonal-Pitch-Class (TPC) chords, Pitch-Class (PC) chords, and Induced Categories (IC). The corresponding grammar models are described in detail in Chapter 8. The most important properties of the models are that 1) the TPC-chord and PC-chord grammars use chord symbols as nonterminals, 2) the fully unsupervised grammar learns nonterminals from the data that do not correspond to chord symbols 1-to-1, and 3) the probability parameterization of the PC-chord grammar is transpositionally invariant. As for the supervised experiments, symmetric Dirichlet distributions are chosen as uninformative priors for the harmony grammar's rewrite-rule distributions. The corresponding hyperparameters were set to $\alpha_r^f = 0.1$ to encode a slight preference for low entropy. Fully unsupervised grammar models that use induced categories are tested, because they constitute the original grammar induction task from computational linguistics. Such models are, however, expected to perform rather poorly.

Additionally, to allow for both left-headed and right-headed rules (either-headed rules, EH), two restricted versions of each harmony grammar were tested that allow for either only Left-Headed rules (LH) or only Right-Headed rules (RH). In the following, the corresponding grammars are called *strictly left-headed* and *strictly right-headed*, respectively. Duplication rules are included in all grammar models. The restricted grammars are tested, because they significantly reduce the amount of possible grammars, which reduces the complexity of the learning task. Furthermore, grammar models that only allow for right-headed rules are hypothesized to perform as well as unrestricted grammar models, because right-headed rules encode goal-directedness. This hypothesis stems from the characterization of functional harmony in music theory. Indeed, Chapter 11 presents computational experiments in which the goal-directedness of functional harmony in Jazz is inferred from chord sequences. The results of this experiment already indicate a weak tendency of right-headedness as shown below.

	Tree Accuracy			Dependency Accuracy			Mean Log Predictive			Tree Height		
Model	EH	LH	RH	EH	LH	RH	EH	LH	RH	EH	LH	RH
TPC	36.80	29.59	43.04	38.26	16.00	55.82	-93.97	-90.08	-86.67	13.64	14.09	12.54
PC	42.37	32.68	**45.56**	46.00	18.68	**58.67**	-85.43	-86.37	**-83.02**	13.29	14.35	12.27
IC	18.73	25.35	26.16	16.54	36.48	27.88	-112.50	-107.11	-109.79	16.36	16.55	15.13
R	57.62	–	–	–	–	–	-40.29	–	–	6.49	–	–
TPC-R	55.79	51.07	57.47	46.31	23.27	64.48	–	–	–	6.76	7.25	7.41
PC-R	56.59	54.28	**57.95**	53.37	23.73	**64.55**	–	–	–	6.58	6.86	6.86
IC-R	57.53	57.00	57.34	36.23	42.16	39.86	–	–	–	6.51	6.50	6.51

Table 10.1 – Unsupervised-grammar-learning results calculated on the test set. Mean accuracies, mean log predictive (MLP), and mean tree height are reported for each grammar model. Tree accuracy and dependency accuracy are reported in percent. The best values are indicated in bold font for each model group. The grammar models vary along three dimensions: 1) harmonic nonterminal representation, 2) headedness restriction of binary rules, and 3) jointly modeling rhythm or not. The tested harmonic nonterminal representations are tonal pitch-class chords (TPC), pitch-class chords (PC), and induced categories (IC). If the headedness is restricted, it allows either only for left-headedness (LH) or right-headedness (RH). If headedness is not restricted, the grammar is called either-headed (EH). Grammars which model rhythm are indicated with the letter R. For instance, TPC-R is the product of the TPC-chord grammar and the rhythm grammar and R is the single-component grammar for rhythm. Note that despite the fact that the rules of the single-component grammar for rhythm are not headed, the measures for that grammar are shown in the either-headed column.

All grammar models for harmony are tested once with jointly modeling rhythm (indicated by the capital letter R) and once as single-component models. Jointly modeling harmony and rhythm is formalized using the product grammar construction; it is expected to significantly improve grammar induction. The single-component model for rhythm is reported for comparison.

10.2.1 Quantitative results and discussion

The results of Experiment A are summarized in Table 10.1. Average tree accuracy, dependency accuracy, MLP, and tree height are reported for each grammar model and each headedness restriction. As in the supervised-learning experiments, jointly modeling rhythm improves the performance of all harmony grammars according to both accuracy measures. The heights of the tree predictions indicate that jointly modeling rhythm leads to more balanced tree predictions.

According to all three evaluation measures, the strictly right-headed PC-chord grammar (PC-RH) is the best single-component grammar for harmony. This is as expected since the PC-chord grammar was the best single-component model for harmony in the supervised-learning experiments and the rules used to analyze the treebank have a strong bias towards right-headedness. About 28% of the binary rules which constitute the treebank are duplications, 70% are right-headed, and 2% are left-headed rules. The most frequent rules of the treebank are shown in Figures 4.8 and 4.9.

Interestingly, the strictly-right headed variants of the single-component PC-chord and TPC-chord grammars perform even better than their corresponding either-headed variants. This finding is remarkable, because all rules of strictly right-headed grammars are also contained in their respective either-headed grammar models. It suggests that the inference procedure gets stuck in local optima for the either-headed models, which is plausible because the space of possible either-headed grammars is much larger than the space of strictly right-headed grammars. Therefore, right-headedness, which encodes the goal-directedness of functional harmony, is shown to be advantageous prior knowledge. The either-headed model performs worse than the strictly right-headed model, because it has to learn the right-headed bias from the data. The performance differences in tree accuracy become smaller, however, when rhythm is jointly modeled.

The expected rule-type proportions for the either-headed grammar models that use chord symbols as nonterminals are shown in Table 10.2. Right-headed rules are between 1.2 and 1.6 times more likely than left-headed rules. These results indicate a weak but general tendency for right-headedness. One reason why this tendency is weak is that the grammar models do not explicitly represent the concept of headedness; the grammar models only allow for headed rules and duplication rules, but the relation between two right-headed rules is a priori the same as the relation between a left-headed and a right-headed rule. Headedness induction is studied in greater detail in Chapter 11.

The grammar models that use induced categories as nonterminals (IC and IC-R) perform overall much worse than the corresponding models that use chord symbols as nonterminals. To interpret the categories learned during inference, Table 10.3 shows the 10 most common chord-symbol terminations for each category of the strictly right-headed product grammar (IC-R-RH). The terminations suggest that the induced categories neither correspond to scale degrees nor to more general chord functions such as tonic or dominant. Instead, they are rather loosely related to keys. For example, the three most common chords to which the category 4 terminates are the scale degrees II, V, and I of a B♭ major key and category 3 can be understood as a mixture of the keys D minor and C minor. Figure10.1 shows the derivation tree of *Summertime*, predicted by the strictly right-headed product grammar that uses induced categories (IC-R-RH). Compared to derivation trees predicted by grammar models that use chord symbols as nonterminals, this tree uses more duplication rules. The increase of duplication rules supports the interpretation of induced categories as keys, because keys change slower than chord functions.

The similarity in tree accuracy between the PC-chord product grammar and the single-component grammar for rhythm suggests that inside the product grammar, the component grammar for harmony is often overruled by the component grammar for rhythm. This is indeed the case as shown in the qualitative analyses presented in the next section.

model	duplications	left-headed rules	right-headed rules
TPC	21.77	35.22	43.02
PC	26.70	28.08	45.22
TPC-R	25.00	33.60	41.41
PC-R	27.91	32.68	39.41

Table 10.2 – Expected rule-type proportions for either-headed grammar models. All values are shown in percent.

category										
1	C7 (25.48)	D7 (12.75)	C6 (12.74)	F6 (9.75)	G7 (9.07)	F^7 (7.79)	A7 (5.71)	Gm (4.29)	G^(2.18)	F7 (2.05)
2	C7 (18.32)	Eb7 (16.77)	G7 (11.83)	F7 (9.97)	Bb7 (7.78)	D7 (4.99)	Ab6 (4.59)	Fm (4.21)	Cm7 (3.85)	Bo7 (2.82)
3	A7 (25.73)	E%7 (12.76)	D%7 (11.60)	G7 (10.05)	Dm7 (9.91)	Db7 (7.36)	Gb7 (3.06)	C#m7 (2.95)	Abm7 (2.86)	D^7 (2.70)
4	Cm7 (33.38)	F7 (21.82)	Bb^7 (14.06)	F^7 (5.47)	F^(5.31)	Am7 (3.78)	Abo7 (2.94)	Fm7 (2.38)	B^7 (1.57)	Bb7 (1.44)
5	Gm7 (30.93)	F^7 (14.84)	Eb7 (9.90)	Ab^7 (9.84)	Bbm7 (7.61)	Db^7 (6.71)	Em7 (5.82)	Cm (3.81)	C^7 (3.76)	D7 (2.82)
6	Fm7 (24.68)	Bb7 (23.35)	Db7 (11.79)	C^(8.66)	C7 (6.82)	Dm (6.43)	A7 (6.06)	Ab^7 (4.40)	Eb^7 (1.48)	C#o7 (1.39)
7	Bbm7 (15.14)	Fm7 (11.86)	G%7 (11.69)	Eb^7 (10.43)	Bb6 (9.61)	Bb^7 (8.86)	Ab^7 (6.26)	Dm (6.05)	Eb7 (4.81)	Gm7 (2.85)
8	Dm7 (27.68)	G7 (23.19)	C^7 (15.33)	Fm6 (7.83)	G^7 (7.39)	F^7 (5.79)	A%7 (4.35)	D7 (2.08)	Bm7 (2.04)	Ebo7 (0.94)
9	Ab7 (19.11)	Em7 (12.50)	B7 (12.50)	Ebm7 (10.87)	B%7 (10.12)	Eb^(6.56)	Gm6 (4.44)	C%7 (3.81)	F#m7 (3.60)	F#%7 (2.66)
10	D7 (26.91)	Am7 (19.12)	E7 (16.45)	Cm6 (9.25)	G6 (7.86)	G7 (5.53)	Bb6 (4.08)	C7 (1.30)	F7 (1.23)	Cm^7, 1.09)

Table 10.3 – Most probable chord-symbol terminals for each induced category of the strictly right-headed product grammar (IC-R-RH). The probability of each chord is shown in percent.

10.2.2 Qualitative analysis of tree predictions

Figures 10.2, 10.3, and 10.4, show the derivation trees of the Jazz standard *Summertime*, predicted by the strictly right-headed (PC-RH), strictly left-headed (PC-LH), and either-headed (PC-EH) versions of the single-component PC-chord grammar model, respectively. The tree prediction of the PC-RH model, shown in Figure 10.2, is similar to the prediction of the PC-EH model, shown in Figure 10.4, and to that of the supervised PC-chord grammar shown in Figure 9.11. The similarity of the predictions by the PC-RH model and its corresponding supervised grammar is explained by the fact that most of the supervised harmony grammars' rules are right-headed. Analogous statements hold for the TPC-chord grammar models. All tree predictions of all models can be found online.[1]

The similarity of the PC-RH and PC-EH predictions reflects that the either-headed grammar variant prefers to use more right-headed than left-headed rules as shown in Table 10.2 and discussed above. The tree predicted by the PC-LH model, shown in Figure 10.3, is higher than the tree predicted by the PC-RH model. It also has some bias for right branchings. A music-theoretical interpretation of left-headed right-branchings is, however, not straight-forward.

Figure 10.5 shows the tree prediction of the strictly right-headed PC-chord product grammar (PC-R-RH). The only split ratio used in that tree is 1/2. This indicates that the rhythm grammar overuses the simplest split ratio 1/2 and futhermore overrules the harmony grammar. The same can be observed for the tree predictions of the other treebank tunes. The constant overruling of the harmony grammar by the rhythm grammar implies that successive grammar induction requires better balancing of the product grammar components, which is discussed in Experiment B presented in the next section.

[1] https://github.com/dharasim/LearnabilityJazzGrammar

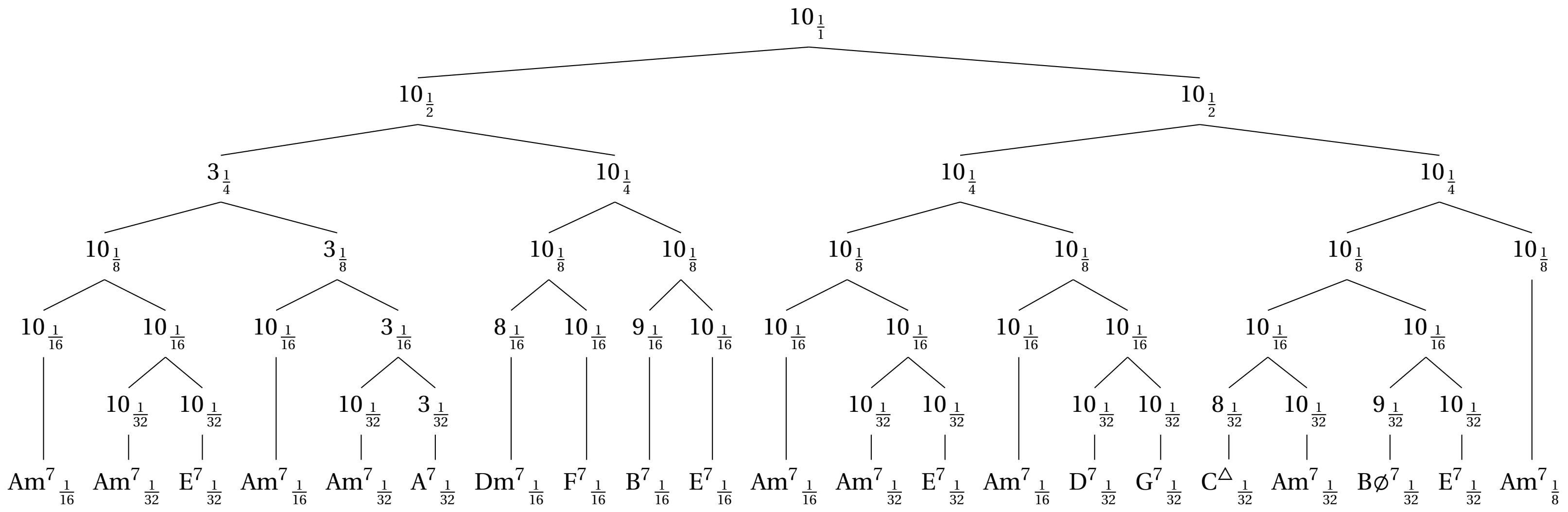

$10_{\frac{1}{1}}$
$10_{\frac{1}{2}}$ $10_{\frac{1}{2}}$
$3_{\frac{1}{4}}$ $10_{\frac{1}{4}}$ $10_{\frac{1}{4}}$ $10_{\frac{1}{4}}$
$10_{\frac{1}{8}}$ $3_{\frac{1}{8}}$ $10_{\frac{1}{8}}$ $10_{\frac{1}{8}}$ $10_{\frac{1}{8}}$ $10_{\frac{1}{8}}$ $10_{\frac{1}{8}}$ $10_{\frac{1}{8}}$
$10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $3_{\frac{1}{16}}$ $8_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $9_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$ $10_{\frac{1}{16}}$
$10_{\frac{1}{32}}$ $10_{\frac{1}{32}}$ $10_{\frac{1}{32}}$ $3_{\frac{1}{32}}$ $10_{\frac{1}{32}}$ $10_{\frac{1}{32}}$ $10_{\frac{1}{32}}$ $10_{\frac{1}{32}}$ $8_{\frac{1}{32}}$ $10_{\frac{1}{32}}$ $9_{\frac{1}{32}}$ $10_{\frac{1}{32}}$
$Am^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{32}}$ $E^7_{\frac{1}{32}}$ $Am^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{32}}$ $A^7_{\frac{1}{32}}$ $Dm^7_{\frac{1}{16}}$ $F^7_{\frac{1}{16}}$ $B^7_{\frac{1}{16}}$ $E^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{16}}$ $Am^7_{\frac{1}{32}}$ $E^7_{\frac{1}{32}}$ $Am^7_{\frac{1}{16}}$ $D^7_{\frac{1}{32}}$ $G^7_{\frac{1}{32}}$ $C^{\triangle}_{\frac{1}{32}}$ $Am^7_{\frac{1}{32}}$ $B\varnothing^7_{\frac{1}{32}}$ $E^7_{\frac{1}{32}}$ $Am^7_{\frac{1}{8}}$

Figure 10.1 – Tree prediction of the strictly right-headed product grammar which uses induced categories (jointly modeling rhythm, unsupervised) for the tune *Summertime*.

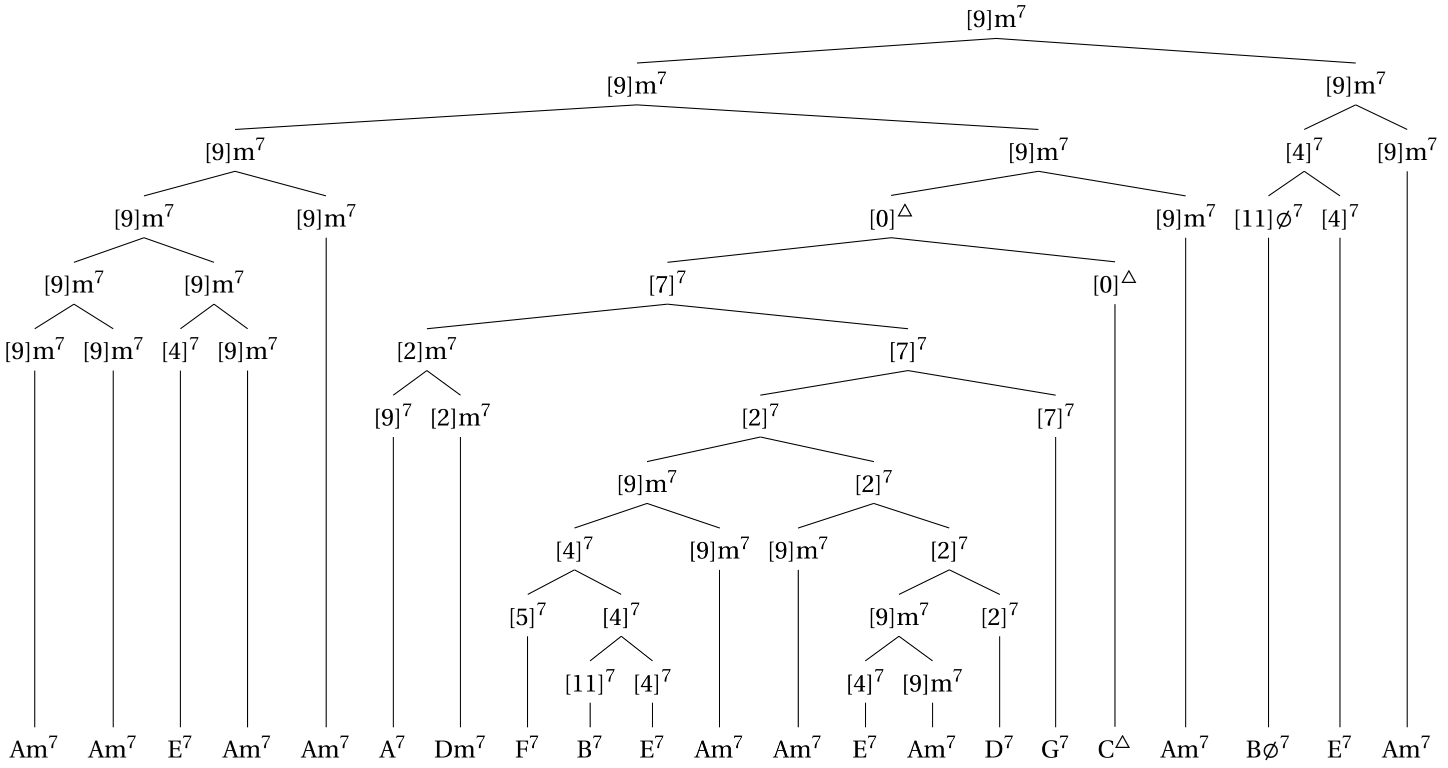


Figure 10.2 – Tree prediction of the strictly right-headed PC-chord grammar (single-component, unsupervised) for the tune *Summertime*.

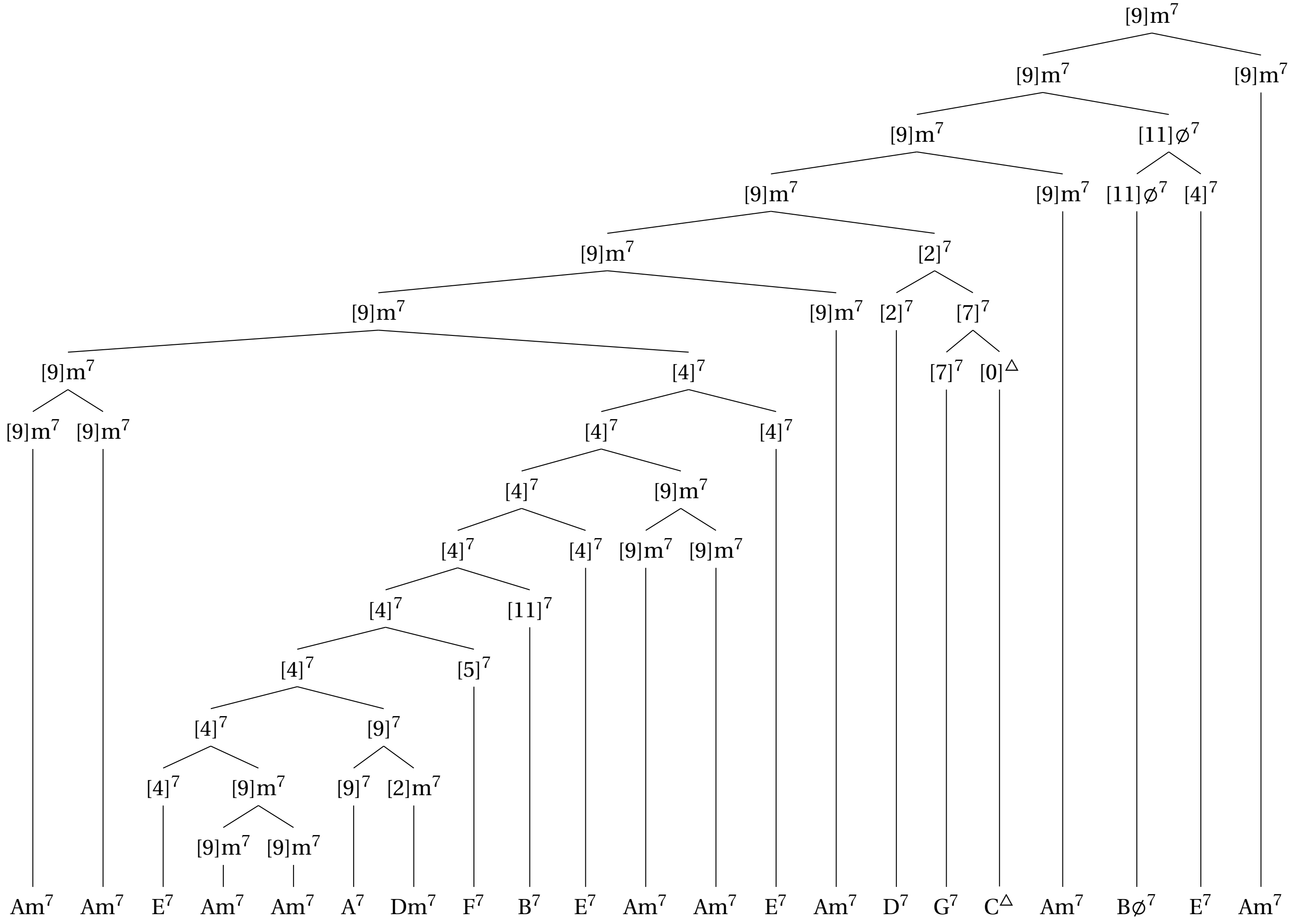

[9]m^7
[9]m^7 [9]m^7
[9]m^7 [11]ø^7
[9]m^7 [9]m^7 [11]ø^7 [4]^7
[9]m^7 [2]^7
[9]m^7 [9]m^7 [2]^7 [7]^7
[9]m^7 [4]^7 [7]^7 [0]$^\triangle$
[9]m^7 [9]m^7 [4]^7 [4]^7
[4]^7 [9]m^7
[4]^7 [4]^7 [9]m^7 [9]m^7
[4]^7 [11]^7
[4]^7 [5]^7
[4]^7 [9]^7
[4]^7 [9]m^7 [9]^7 [2]m^7
[9]m^7 [9]m^7
Am^7 Am^7 E^7 Am^7 Am^7 A^7 Dm^7 F^7 B^7 E^7 Am^7 Am^7 E^7 Am^7 D^7 G^7 C$^\triangle$ Am^7 Bø^7 E^7 Am^7

Figure 10.3 – Tree prediction of the strictly left-headed PC-chord grammar (single-component, unsupervised) for the tune *Summertime*.

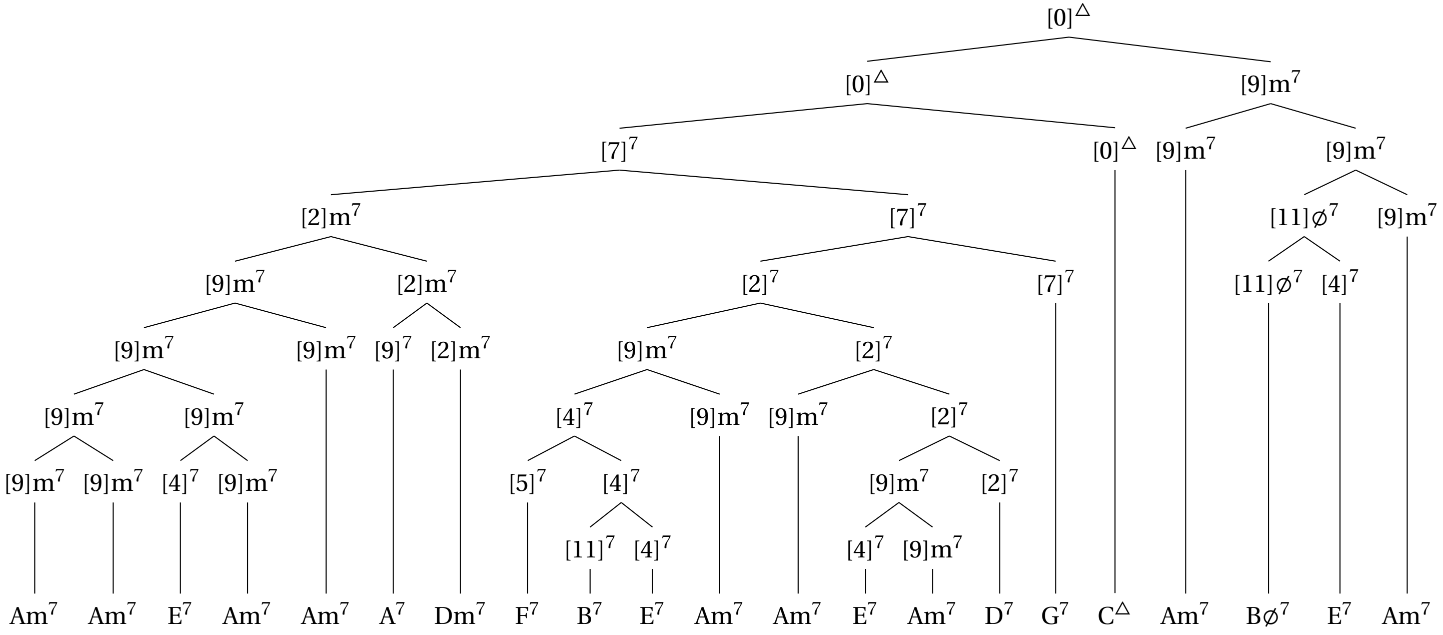


Figure 10.4 – Tree prediction of the either-headed PC-chord grammar (single-component, unsupervised) for the tune *Summertime*.

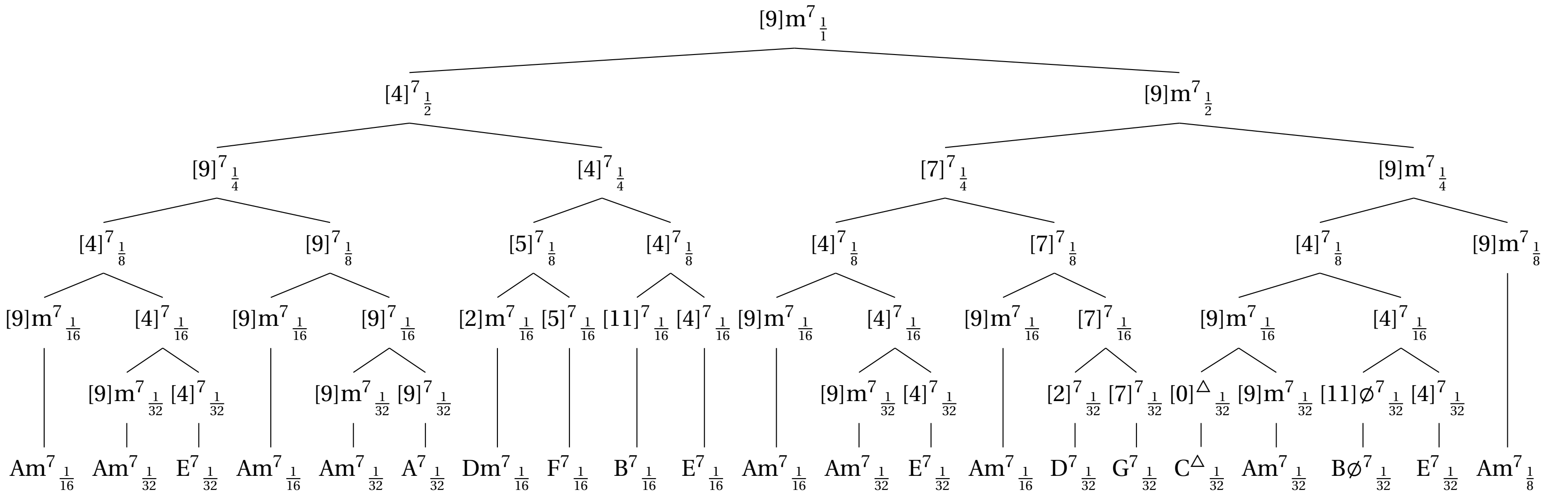

$[9]\mathrm{m}^7_{\frac{1}{1}}$
$[4]^7_{\frac{1}{2}}$ $[9]\mathrm{m}^7_{\frac{1}{2}}$
$[9]^7_{\frac{1}{4}}$ $[4]^7_{\frac{1}{4}}$ $[7]^7_{\frac{1}{4}}$ $[9]\mathrm{m}^7_{\frac{1}{4}}$
$[4]^7_{\frac{1}{8}}$ $[9]^7_{\frac{1}{8}}$ $[5]^7_{\frac{1}{8}}$ $[4]^7_{\frac{1}{8}}$ $[4]^7_{\frac{1}{8}}$ $[7]^7_{\frac{1}{8}}$ $[4]^7_{\frac{1}{8}}$ $[9]\mathrm{m}^7_{\frac{1}{8}}$
$[9]\mathrm{m}^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{16}}$ $[9]^7_{\frac{1}{16}}$ $[2]\mathrm{m}^7_{\frac{1}{16}}$ $[5]^7_{\frac{1}{16}}$ $[11]^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{16}}$ $[7]^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{16}}$
$[9]\mathrm{m}^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$ $[9]\mathrm{m}^7_{\frac{1}{32}}$ $[9]^7_{\frac{1}{32}}$ $[9]\mathrm{m}^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$ $[2]^7_{\frac{1}{32}}$ $[7]^7_{\frac{1}{32}}$ $[0]^{\triangle}_{\frac{1}{32}}$ $[9]\mathrm{m}^7_{\frac{1}{32}}$ $[11]\varnothing^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$
$\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{E}^7_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{A}^7_{\frac{1}{32}}$ $\mathrm{Dm}^7_{\frac{1}{16}}$ $\mathrm{F}^7_{\frac{1}{16}}$ $\mathrm{B}^7_{\frac{1}{16}}$ $\mathrm{E}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{E}^7_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{D}^7_{\frac{1}{32}}$ $\mathrm{G}^7_{\frac{1}{32}}$ $\mathrm{C}^{\triangle}_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{B}\varnothing^7_{\frac{1}{32}}$ $\mathrm{E}^7_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{8}}$

Figure 10.5 – Tree prediction of the strictly right-headed PC-chord product grammar (jointly modeling rhythm, unsupervised) for the tune *Summertime*.

10.3 Experiment B: Prior preference for simple duration splits

Experiment B investigates the regularization of the grammar component for rhythm. The goal is to find a prior distribution that 1) increases the entropy of the split-ratio distribution and 2) balances the rhythm-grammar against the harmony-grammar component of the product grammar. The strictly right-headed PC-chord and TPC-chord product grammars are considered in this experiment. The prior for the rhythm component used in Experiment A (the unregularized version of the rhythm model) is compared to two regularized versions. As in Experiment A, the priors are variants of Dirichlet distributions, only the hyperparameters differ between versions. The first regularization symmetrically increases the prior probability of all split ratios. The second regularization increases the prior probabilities of simple split ratios more than that of complex split ratios.

The results show that the prior that favors simple split ratios performs best. A brief qualitative analysis of the regularized grammar's tree predictions confirms that the inferred grammar is of high quality. Despite the fact that the parameters of the regularizations are set by hand in this study, the results do show that a prior preference for simple split ratios is beneficial. What they do not show is that the optimal strength of the regularization can be inferred automatically from the data.

10.3.1 Regularizing priors for the split-ratio distribution

The approach of using prior distributions for regularization is common in Bayesian statistics. In Experiment A, the split-ratio distributions obtained by unregularized learning do not have enough entropy — they put too much probability mass on the single value $\frac{1}{2}$. Therefore, Experiment B applies priors that increase the probabilities of distributions which use many different split-ratios. Since the split ratio $\frac{1}{2}$ is still most common in the derivation trees of the regularized rhythm grammars, the probabilities of the derivation trees for rhythm decrease with increasing entropy of the split ratio distribution. This indirectly increases the influence of the grammar component for harmony on tree predictions, and thus balances the components.

The first regularization uses a symmetric Dirichlet prior like in Experiment A, but increases the concentration parameter from 0.1 to 400. Experiments with different concentration parameters found that the value of 400 works well. The high concentration parameter implies that the split-ratio distribution is likely to distribute the probability mass more equally between the split ratios. However, this leads to rhythm grammars that use unusual duration splits such as $\frac{3}{32}$, as shown below.

The second regularization also increases the entropy of the split-ratio distribution, but does so by favoring simple split ratios. It therefore relies on a mathematically precise definition of what is meant by a simple ratio. This study proposes to use the *Calkin-Wilf tree* (Calkin and Wilf, 2000), a mathematical object from number theory, to measure a ratio's simplicity. The upper levels of the tree are shown in Figure 10.6. The Calkin-Wilf tree is an infinitely large

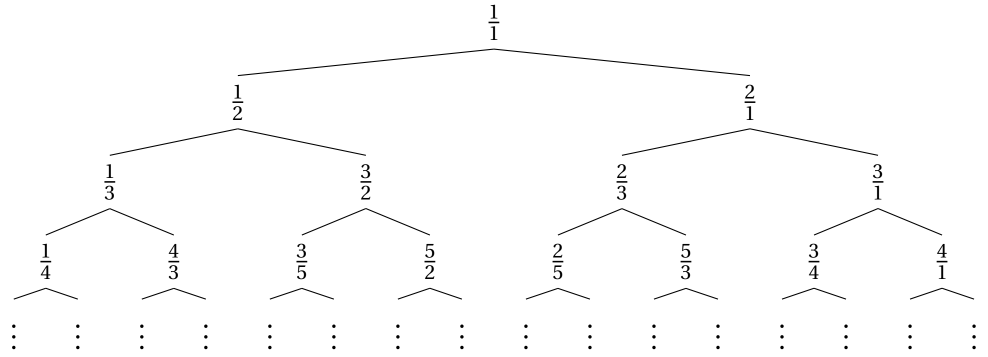


Figure 10.6 – First three levels of the infinitely large Calkin-Wilf tree. All positive rational numbers are represented exactly once. This tree is used to formalize the simplicity of a ratio; the higher a ratio is in the Calkin-Wilf tree the simpler it is.

binary tree that contains each positive rational number exactly once. It is defined recursively as follows: the root of the tree (e.g., the topmost node in Figure 10.6) is the number 1. The two children of a rational number $\frac{a}{b}$, where a and b are positive natural numbers such that their greatest common divisor is 1, are $\frac{a}{a+b}$ and $\frac{a+b}{b}$. The simplicity of a rational number is operationalized as the level on which it occurs in the Calkin-Wilf tree. The level of a ratio s is denoted by $\lambda(s)$, and the level of the root of the tree is set to zero, $\lambda(1) = 0$. For example, $\frac{1}{2}$ is simpler than $\frac{1}{3}$ since $\lambda(\frac{1}{2}) = 1 < 2 = \lambda(\frac{1}{3})$, $\frac{2}{3}$ is simpler than $\frac{3}{5}$ since $\lambda(\frac{2}{3}) = 2 < 3 = \lambda(\frac{3}{5})$, and $\frac{1}{4}$ is as simple as $\frac{3}{5}$ since $\lambda(\frac{1}{4}) = 3 = \lambda(\frac{3}{5})$. This operationalization is certainly not the only reasonable one, but one based on a well-established mathematical object. The Calkin-Wilf tree is closely related to but simpler than the more popular Stern-Brocot tree (Stern, 1858; Brocot, 1860). Since the levels on which the numbers occur are equal for both trees, they are equivalent for the purpose of this study.

The Calkin-Wilf prior over split-ratio distributions is based on a Dirichlet distribution. The parameter of that Dirichlet distribution is a vector that assigns each split ratio a pseudocount as described in Section 7.2 following Equation 7.19. The pseudocount of a split ratio is proportional to the expected prior probability of that split ratio. Denote the pseudocount vector by $\boldsymbol{\alpha}$ and the pseudocount of a split ratio $0 < s < 1$ by $\boldsymbol{\alpha}_s$. The Calkin-Wilf prior sets the pseudocounts to

$$\alpha_s = D(L - \lambda(s)) \tag{10.7}$$

where L is the maximal level of the split ratios allowed by the grammar and D is a free parameter (a positive integer). The parameter D describes the pseudocount difference between the levels, for example

$$\boldsymbol{\alpha}_{\frac{1}{2}} - \boldsymbol{\alpha}_{\frac{1}{3}} = DL - D\lambda\left(\frac{1}{2}\right) - DL + D\lambda\left(\frac{1}{3}\right) = -D \cdot 1 + D \cdot 2 = D. \tag{10.8}$$

The maximal level $L = 8$ was chosen, which allows the split ratio $\frac{1}{8}$. The difference parameter was set to $D = 300$, which was found to work best. This choice leads for example to $\boldsymbol{\alpha}_{\frac{1}{2}} = 2100$, $\boldsymbol{\alpha}_{\frac{1}{3}} = 1800$, and $\boldsymbol{\alpha}_{\frac{1}{8}} = 300$.

10.3.2 Results and discussion

The results of Experiment B are summarized in Table 10.4. According to tree accuracy and dependency accuracy, the Calkin-Wilf regularization performs best, followed by the symmetric regularization, which is still slightly better than no regularization. The PC-chord model can benefit more from the regularization than the TPC-chord model and achieves a tree accuracy similar to that from the supervised experiments (see Table 9.1). The average heights of the tree predictions increase slightly under the regularization. This makes sense, because the tree predictions of the unregularized versions are almost maximally balanced. The MLPs of the PC-chord model are better than the MLPs of the TPC-chord model for all regularization variants, confirming the positive effect of a transpositional invariant parameterization. The MLPs are not comparable across regularization variants, because strong regularization generally decreases predictive probabilities.

The tree predictions of the tune *Summertime* are qualitatively compared in this paragraph for the PC-chord model. The results are analogous for the TPC model. Figure 10.7 and Figure 10.8 show the tree predictions of the PC-chord product grammar with symmetric regularization and Calkin-Wilf regularization, respectively. The symmetric regularization leads to unusual split ratios such as $\frac{3}{32}$, $\frac{4}{7}$, and $\frac{1}{29}$. The corresponding tree prediction is not able to identify much of the hierarchical phrase structure of the tune. The tree prediction is not better than the prediction of the single-component grammar for harmony shown in Figure 10.2. In contrast, the prediction of the grammar inferred with the Calkin-Wilf regularization is much better. Except for the unconventional open constituent which closes on Dm^7, the tree describes the harmonic and formal structure of the tune well. Similar observations were made for the tree predictions of the other treebank tunes which can be found online.[2]

Model	Regularization	Tree Acc.	Dep. Acc.	Tree Height
TPC-R-RH	none	57.47	64.48	7.41
TPC-R-RH	symmetric	58.06	65.80	8.19
TPC-R-RH	Calkin-Wilf	60.23	67.48	7.88
PC-R-RH	none	57.95	64.55	6.86
PC-R-RH	symmetric	59.55	66.46	7.83
PC-R-RH	Calkin-Wilf	62.34	69.61	7.79

Table 10.4 – Effect of duration-split ratio regularization on grammar learning. Symmetric and Calkin-Wilf regularization are tested for two product grammar models, the strictly right-headed TPC-chord product grammar (TPC-R-RH) and the strictly right-headed PC-chord product grammar (PC-R-RH).

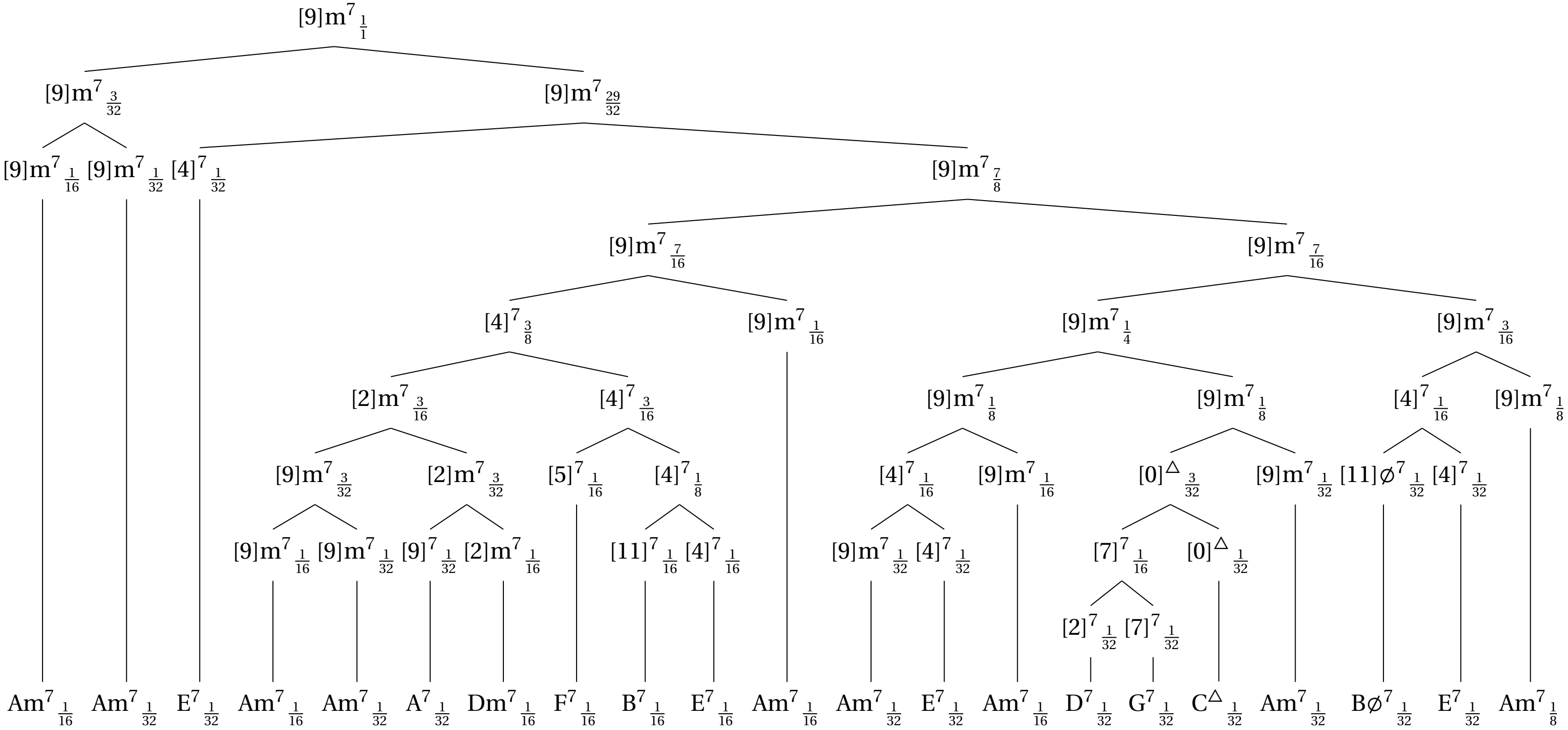

$[9]\mathrm{m}^7_{\frac{1}{1}}$
$[9]\mathrm{m}^7_{\frac{3}{32}}$ $[9]\mathrm{m}^7_{\frac{29}{32}}$
$[9]\mathrm{m}^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$ $[9]\mathrm{m}^7_{\frac{7}{8}}$
$[9]\mathrm{m}^7_{\frac{7}{16}}$ $[9]\mathrm{m}^7_{\frac{7}{16}}$
$[4]^7_{\frac{3}{8}}$ $[9]\mathrm{m}^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{4}}$ $[9]\mathrm{m}^7_{\frac{3}{16}}$
$[2]\mathrm{m}^7_{\frac{3}{16}}$ $[4]^7_{\frac{3}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{8}}$ $[9]\mathrm{m}^7_{\frac{1}{8}}$ $[4]^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{8}}$
$[9]\mathrm{m}^7_{\frac{3}{32}}$ $[2]\mathrm{m}^7_{\frac{3}{32}}$ $[5]^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{8}}$ $[4]^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{16}}$ $[0]^{\triangle}_{\frac{3}{32}}$ $[9]\mathrm{m}^7_{\frac{1}{32}}$ $[11]\varnothing^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$
$[9]\mathrm{m}^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{32}}$ $[9]^7_{\frac{1}{32}}$ $[2]\mathrm{m}^7_{\frac{1}{16}}$ $[11]^7_{\frac{1}{16}}$ $[4]^7_{\frac{1}{16}}$ $[9]\mathrm{m}^7_{\frac{1}{32}}$ $[4]^7_{\frac{1}{32}}$ $[7]^7_{\frac{1}{16}}$ $[0]^{\triangle}_{\frac{1}{32}}$
$[2]^7_{\frac{1}{32}}$ $[7]^7_{\frac{1}{32}}$
$\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{E}^7_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{A}^7_{\frac{1}{32}}$ $\mathrm{Dm}^7_{\frac{1}{16}}$ $\mathrm{F}^7_{\frac{1}{16}}$ $\mathrm{B}^7_{\frac{1}{16}}$ $\mathrm{E}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{E}^7_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{16}}$ $\mathrm{D}^7_{\frac{1}{32}}$ $\mathrm{G}^7_{\frac{1}{32}}$ $\mathrm{C}^{\triangle}_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{32}}$ $\mathrm{B}\varnothing^7_{\frac{1}{32}}$ $\mathrm{E}^7_{\frac{1}{32}}$ $\mathrm{Am}^7_{\frac{1}{8}}$

Figure 10.7 – Tree prediction of the strictly right-headed PC-chord product grammar with symmetric regularization of duration-split ratios (jointly modeling rhythm, unsupervised) for the tune *Summertime*.

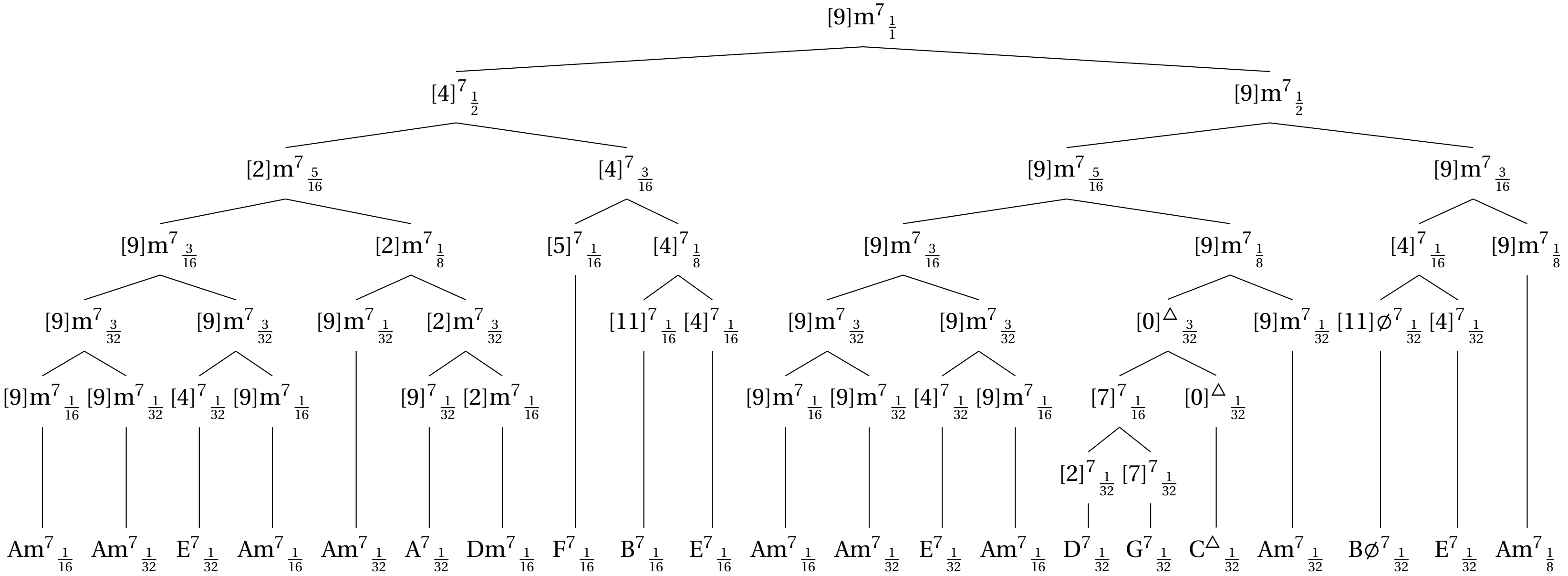

$[9]\mathrm{m}^7_{\frac{1}{1}}$
$[4]^7_{\frac{1}{2}}$
$[9]\mathrm{m}^7_{\frac{1}{2}}$
$[2]\mathrm{m}^7_{\frac{5}{16}}$
$[4]^7_{\frac{3}{16}}$
$[9]\mathrm{m}^7_{\frac{5}{16}}$
$[9]\mathrm{m}^7_{\frac{3}{16}}$
$[9]\mathrm{m}^7_{\frac{3}{16}}$
$[2]\mathrm{m}^7_{\frac{1}{8}}$
$[5]^7_{\frac{1}{16}}$
$[4]^7_{\frac{1}{8}}$
$[9]\mathrm{m}^7_{\frac{3}{16}}$
$[9]\mathrm{m}^7_{\frac{1}{8}}$
$[4]^7_{\frac{1}{16}}$
$[9]\mathrm{m}^7_{\frac{1}{8}}$
$[9]\mathrm{m}^7_{\frac{3}{32}}$
$[9]\mathrm{m}^7_{\frac{3}{32}}$
$[9]\mathrm{m}^7_{\frac{1}{32}}$
$[2]\mathrm{m}^7_{\frac{3}{32}}$
$[11]^7_{\frac{1}{16}}$
$[4]^7_{\frac{1}{16}}$
$[9]\mathrm{m}^7_{\frac{3}{32}}$
$[9]\mathrm{m}^7_{\frac{3}{32}}$
$[0]^{\triangle}_{\frac{3}{32}}$
$[9]\mathrm{m}^7_{\frac{1}{32}}$
$[11]\varnothing^7_{\frac{1}{32}}$
$[4]^7_{\frac{1}{32}}$
$[9]\mathrm{m}^7_{\frac{1}{16}}$
$[9]\mathrm{m}^7_{\frac{1}{32}}$
$[4]^7_{\frac{1}{32}}$
$[9]\mathrm{m}^7_{\frac{1}{16}}$
$[9]^7_{\frac{1}{32}}$
$[2]\mathrm{m}^7_{\frac{1}{16}}$
$[9]\mathrm{m}^7_{\frac{1}{16}}$
$[9]\mathrm{m}^7_{\frac{1}{32}}$
$[4]^7_{\frac{1}{32}}$
$[9]\mathrm{m}^7_{\frac{1}{16}}$
$[7]^7_{\frac{1}{16}}$
$[0]^{\triangle}_{\frac{1}{32}}$
$[2]^7_{\frac{1}{32}}$
$[7]^7_{\frac{1}{32}}$
$\mathrm{Am}^7_{\frac{1}{16}}$
$\mathrm{Am}^7_{\frac{1}{32}}$
$\mathrm{E}^7_{\frac{1}{32}}$
$\mathrm{Am}^7_{\frac{1}{16}}$
$\mathrm{Am}^7_{\frac{1}{32}}$
$\mathrm{A}^7_{\frac{1}{32}}$
$\mathrm{Dm}^7_{\frac{1}{16}}$
$\mathrm{F}^7_{\frac{1}{16}}$
$\mathrm{B}^7_{\frac{1}{16}}$
$\mathrm{E}^7_{\frac{1}{16}}$
$\mathrm{Am}^7_{\frac{1}{16}}$
$\mathrm{Am}^7_{\frac{1}{32}}$
$\mathrm{E}^7_{\frac{1}{32}}$
$\mathrm{Am}^7_{\frac{1}{16}}$
$\mathrm{D}^7_{\frac{1}{32}}$
$\mathrm{G}^7_{\frac{1}{32}}$
$\mathrm{C}^{\triangle}_{\frac{1}{32}}$
$\mathrm{Am}^7_{\frac{1}{32}}$
$\mathrm{B}\varnothing^7_{\frac{1}{32}}$
$\mathrm{E}^7_{\frac{1}{32}}$
$\mathrm{Am}^7_{\frac{1}{8}}$

Figure 10.8 – Tree prediction of the strictly right-headed PC-chord product grammar with Calkin-Wilf regularization of duration-split ratios (jointly modeling rhythm, unsupervised) for the tune *Summertime*.

10.3.3 Comparison to supervised grammar models

Table 10.5 compares the best unsupervised grammar model (PC-R-RH with Calkin-Wilf regularization) to the best supervised model as described in the previous chapter (PC-R). The tree accuracies of the models differ only by about one percent. However, the supervised model achieves an about five percent higher dependency accuracy. The tree predictions of the supervised grammar are on average higher than those of the unsupervised grammar, but they are both close to the average height of the treebank trees which amounts to 7.5. This is explained by the facts that rhythmic regularization generally increases the height of the tree predictions and that the supervised grammar models are not regularized.

With respect to tree accuracy, the performance of the best unsupervised model is similar to that of the best supervised model. This is confirmed by qualitative analyses of the tree predictions. In computational linguistics, it is unusual that unsupervised grammar models perform as similarly as supervised models (e.g., Kim et al., 2019). The similarity obtained in this thesis can be explained by two main reasons for why the supervised model does not perform better. The first reason is that state-of-the-art parsers for natural language are much more sophisticated than the supervised grammar models presented in Capter 9. Since the focus of this thesis lies on grammar induction, the supervised grammar models were chosen to be as similar as possible to the unsupervised models in order to facilitate direct comparisons. Because of the simple structure of the supervised models, there is room for improvement expected for future research. Furthermore, the parameters of the rhythmic regularizations were set by hand in this thesis. The unsupervised models most probably do not perform so well when they have to learn those parameters from data. Also, the supervised models are expected to perform better with a rhythmic regularization. The second reason is that other factors such as melody and meter must be taken into account in order to accurately describe harmonic dependency structures. Without that additional information, chord sequences might be too ambiguous to accurately predict the treebank trees.

Model	Tree Acc.	Dep. Acc.	Tree Height
best unsupervised	62.3	69.6	7.8
best supervised	63.1	74.2	7.3

Table 10.5 – Comparison of the best unsupervised grammar model (PC-R-RH with Calkin-Wilf regularization) to the best supervised model (PC-R). Both grammars jointly model harmony and rhythm.

11 The learnability of goal-directedness

Functional harmony as presented in Chapter 1 postulates the goal-directedness of harmonic dependencies that is represented in formal grammar models by right-headed rules. The unsupervised experiments reported in the previous chapter show that the prior assumption of goal-directedness improves the learning capability of grammar models for harmony. Furthermore, grammars models (called either-headed) that could freely chose from left- and right-headed rules to represent syntactic structure learned a slight tendency for right-headedness. However, the facts that strictly right-headed grammars perform at least as well as either-headed grammars and that left-headed grammars perform significantly worse stand in discrepancy to the result that the either either-headed grammar models did not learn to use more right-headed rules. One reason is that the space of all either-headed grammars is very large; almost each grammar rule corresponds to one dimension. The grammar space is therefore likely to contain various local optima. Since either-headed grammar models do not represent the abstract concept of headedness, they do not distinguish grammars that use as much left- as right-headed rules from grammars that use only one headedness direction. Intuitively, a grammar is simpler if it only contains (mostly) one headedness direction. Applying Occam's razor (see e.g., MacKay, 2003, chapter 28), such a grammar should be preferred over a grammar with balanced headedness directions if both grammars perform equally well.

This chapter presents a computational grammar model that explicitly represents the abstract concept of headedness and induces the headedness proportions from Jazz chord sequences. To only encode domain-general prior knowledge, the model is based on the TPC-chord grammar without jointly modeling rhythm. The results show that the induced grammar uses only right-headed rules. The goal-directedness of functional harmony is thus demonstrated to be learnable without any music-specific prior knowledge.

A second experiment using artificial data is used to verify the correct functionality of the headedness induction. In that experiment, the model was able to correctly distinguish various degrees of left- and right-headedness from sequential datasets that were automatically generated using grammars whose headedness usage was set by hand.

11.1 A grammar model for headedness induction

The grammar model for headedness induction is based on the TPC-chord grammar model described in Section 8.1.1. A TPC-chord grammar is a probabilistic context-free grammar that uses chord symbols both as terminals and nonterminals. The chord symbols are characterized by a root represented by a Tonal-Pitch Class (TPC) and a chord form represented by a string. The difference between TPC-chord grammar models used before and the headedness-induction model presented in this chapter is the parameterization of the rewrite probabilities.

This paragraph briefly summarizes the notation of rewrite-rule probabilities. All nonterminals that are projected onto the same feature $f \in \Phi$ under the nonterminal feature projection $\phi\colon N \to \Phi$ (where Φ denotes the set of all feature instantiations) share a distribution $\boldsymbol{\theta}^f$ over rewrite probabilities (as described in Section 7.1). The probability that a rule $r \in R$ rewrites a nonterminal with feature f is denoted by $\boldsymbol{\theta}^f_r$. The collection of all rewrite-rule distributions is denoted by $\boldsymbol{\theta}$. For the TPC-chord grammar, the set of feature instantiations equals the set of nonterminals, $\Phi = N$, and the feature projection is the identity. In the following, however, the model is presented with respect to an arbitrary feature projection to show that the general idea of the headedness induction is applicable to arbitrary PACFGs. This also achieves a notation consistent with the rest of this thesis.

Previous grammar models directly parameterized the rewrite-rule distributions $\boldsymbol{\theta}^f$ as independent Dirichlet distributions (see Section 7.2). In contrast, the idea of the headedness-induction model is to link the probabilities of rewrite rules that have a common rule type. In the experiments, the set of rule types consists of values for unary rules, duplications, left-headed, and right-headed rules. The linked probabilities then enable the model to abstractly learn about rule types in general and headedness in particular.

The set of rule types is denoted by Ψ, and the function that maps a rule $r \in R$ to its rule type $\psi(r)$ is denoted by $\psi\colon R \to \Psi$. The process of sampling a rule r to rewrite a nonterminal with feature f is a two-step generation process. First, a rule type $\rho \in \Psi$ (e.g., for a left- or right-headed rule) is drawn from a distribution of rule types conditioned on the feature f. That distribution is denoted by $\boldsymbol{\xi}^f$ and the probability that ρ is sampled from $\boldsymbol{\xi}^f$ is denoted by $\boldsymbol{\xi}^f_\rho$. To ensure the consistency of the model, the probability $\boldsymbol{\xi}^f_\rho$ is required to be positive if and only if there is a rule $r \in R_f$ that has type ρ, $\psi(r) = \rho$. In the second step, a rule $r \in R_f$ is drawn from a distribution of rules that have type ρ. That distribution is denoted by $\boldsymbol{\zeta}^{f,\rho}$ and the probability that r is sampled from $\boldsymbol{\zeta}^{f,\rho}$ is denoted by $\boldsymbol{\zeta}^{f,\rho}_r$. By taking the generation step together, the probability of a rule r rewriting a nonterminal with feature f is given as the product

$$\boldsymbol{\theta}^f_r = \boldsymbol{\xi}^f_{\psi(r)}\, \boldsymbol{\zeta}^{f,\psi(r)}_r. \tag{11.1}$$

As an example, consider the process of sampling a rule to rewrite the TPC-chord $\mathrm{C}^{\triangle}$. Since the feature projection of the TPC-chord grammar is the identity, $\phi(\mathrm{C}^{\triangle}) = \mathrm{C}^{\triangle}$. In the first step, a rule type, say RHR for a right-headed rule, is sampled from the distribution of rule types $\boldsymbol{\xi}^{\phi(\mathrm{C}^\triangle)} = \boldsymbol{\xi}^{\mathrm{C}^\triangle}$. The rule type RHR can be sampled, because there are right-headed rules that are applicable to $\mathrm{C}^\triangle$. In the second step, a right-headed rule with $\mathrm{C}^\triangle$ on its left-hand side is sampled from the distribution $\boldsymbol{\zeta}^{\mathrm{C}^\triangle,\mathrm{RHR}}$, say $C^\triangle \longrightarrow G^7\ C^\triangle$. That rule can be sampled, because it has rule type RHR, $\psi(C^\triangle \longrightarrow G^7\ C^\triangle) = \mathrm{RHR}$. The probability of applying the rule $C^\triangle \longrightarrow G^7\ C^\triangle$ to the TPC-chord symbol $C^\triangle$ is the product

$$\boldsymbol{\theta}^{\mathrm{C}^\triangle}_{C^\triangle \longrightarrow G^7\ C^\triangle} = \boldsymbol{\xi}^{\mathrm{C}^\triangle}_{\mathrm{RHR}}\ \boldsymbol{\zeta}^{\mathrm{C}^\triangle,\mathrm{RHR}}_{C^\triangle \longrightarrow G^7\ C^\triangle}. \tag{11.2}$$

Since the rule-type distributions $\boldsymbol{\xi}^f$ and the rule distributions $\boldsymbol{\zeta}^{f,\rho}$ are unknown, they are modeled as random variables. The distribution over rule-type distributions $\boldsymbol{\xi}^f$ is chosen as a Dirichlet distribution with pseudocount vector $\boldsymbol{\mu}^f$, and the distribution over rule distributions $\boldsymbol{\zeta}^{f,\rho}$ is chosen as a Dirichlet distribution with pseudocount vector $\boldsymbol{\nu}^{f,\rho}$. All distributions $\boldsymbol{\xi}^f$ and $\boldsymbol{\zeta}^{f,\rho}$ are furthermore assumed to be independent.

The full probabilistic model is given by the following factorization of the joint distribution of rewrite probabilities $\boldsymbol{\theta}$, derivations $\bar{\boldsymbol{r}}$, and chord sequences $\bar{\boldsymbol{w}}$:

$$\begin{aligned}
p(\boldsymbol{\theta}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{w}}) &= p(\boldsymbol{\theta}) \prod_{i=1}^{I} p(\boldsymbol{w}^i \mid \boldsymbol{r}^i)\ p(\boldsymbol{r}^i \mid \boldsymbol{\theta}) && (11.3)\\
&= p(\boldsymbol{\theta}) \prod_{i=1}^{I} \mathbb{1}\Big(\boldsymbol{r}^i(\mathrm{Start}) = \boldsymbol{w}^i\Big) \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\theta}^{\phi(A^i_k)}_{\boldsymbol{r}^i_k} && (11.4)\\
&= p(\boldsymbol{\xi})\ p(\boldsymbol{\zeta}) \prod_{i=1}^{I} \mathbb{1}\Big(\boldsymbol{r}^i(\mathrm{Start}) = \boldsymbol{w}^i\Big) \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\xi}^{\phi(A^i_k)}_{\psi(\boldsymbol{r}^i_k)}\ \boldsymbol{\zeta}^{\phi(A^i_k),\psi(\boldsymbol{r}^i_k)}_{\boldsymbol{r}^i_k} && (11.5)\\
&= \left(\prod_{f\in\Phi} p(\boldsymbol{\xi}^f)\right)\left(\prod_{f\in\Phi}\prod_{\rho\in\Psi} p(\boldsymbol{\zeta}^{f,\rho})\right) \prod_{i=1}^{I} \mathbb{1}\Big(\boldsymbol{r}^i(\mathrm{Start}) = \boldsymbol{w}^i\Big) \prod_{k=1}^{|\boldsymbol{r}^i|} \boldsymbol{\xi}^{\phi(A^i_k)}_{\psi(\boldsymbol{r}^i_k)}\ \boldsymbol{\zeta}^{\phi(A^i_k),\psi(\boldsymbol{r}^i_k)}_{\boldsymbol{r}^i_k} && (11.6)
\end{aligned}$$

where I denotes the number of chord sequences and A^i_k denotes the leftmost nonterminal of the sequence $\boldsymbol{r}^i_{1:k-1}(\mathrm{Start})$ — the nonterminal to which the rule $\boldsymbol{r}^i_k$ is applied in the derivation of the terminal sequence $\boldsymbol{w}^i$.

The posterior distribution $p(\boldsymbol{\xi}, \boldsymbol{\zeta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}})$ is approximated using Coordinate Ascent Variational Inference (CAVI) analogous to the approximation described in Section 7.4. The posterior is approximated using a distribution which assumes independence of the random variables $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$, and $\bar{\boldsymbol{r}}$,

$$p(\boldsymbol{\xi}, \boldsymbol{\zeta}, \bar{\boldsymbol{r}} \mid \bar{\boldsymbol{w}}) \approx q(\boldsymbol{\xi}, \boldsymbol{\zeta}, \bar{\boldsymbol{r}}) = q(\boldsymbol{\xi})\ q(\boldsymbol{\zeta})\ q(\bar{\boldsymbol{r}}), \tag{11.7}$$

where the variational approximations are in the same class as the conditional distributions of the model, $q(\boldsymbol{\xi}^f) = \mathrm{Dir}(\tilde{\boldsymbol{\mu}}^f)$, $q(\boldsymbol{\zeta}^{f,\rho}) = \mathrm{Dir}(\tilde{\boldsymbol{\nu}}^{f,\rho})$, and $q(\bar{\boldsymbol{r}}) = \prod_{i=1}^{I}\prod_{k=1}^{|\boldsymbol{r}^i|} \tilde{\boldsymbol{\theta}}^{\phi(A^i_k)}_{\boldsymbol{r}^i_k}$. As before, the parameters of the variational approximation are denoted using a tilde.

The CAVI parameter updates that lead to a locally optimal approximation of the posterior are given as:

$$\tilde{\boldsymbol{\mu}}_\rho^f := \boldsymbol{\mu}_\rho^f + \mathbb{E}_{q(\bar{\boldsymbol{r}})} \left[\sum_{i=1}^{I} \sum_{k=1}^{|\boldsymbol{r}^i|} \mathbb{1}\left(f = \phi(A_k^i)\right) \mathbb{1}\left(\rho = \psi(\boldsymbol{r}_k^i)\right) \right] \tag{11.8}$$

$$\tilde{\boldsymbol{v}}_r^{f,\rho} := \boldsymbol{v}_r^{f,\rho} + \mathbb{E}_{q(\bar{\boldsymbol{r}})} \left[\sum_{i=1}^{I} \sum_{k=1}^{|\boldsymbol{r}^i|} \mathbb{1}\left(f = \phi(A_k^i)\right) \mathbb{1}\left(\rho = \psi(\boldsymbol{r}_k^i)\right) \mathbb{1}\left(r = \boldsymbol{r}_k^i\right) \right] \tag{11.9}$$

$$\tilde{\boldsymbol{\theta}}_r^f := \exp \mathbb{E}_{q(\boldsymbol{\theta})} \left[\log \boldsymbol{\theta}_r^f \right] = \frac{\exp \gamma \left(\tilde{\boldsymbol{\mu}}_{\psi(r)}^f \right)}{\exp \gamma \left(\sum_{\rho \in \Psi_f} \tilde{\boldsymbol{\mu}}_\rho^f \right)} \frac{\exp \gamma \left(\tilde{\boldsymbol{v}}_r^{f,\psi(r)} \right)}{\exp \gamma \left(\sum_{r' \in R_f \colon \phi(r') = \phi(r)} \tilde{\boldsymbol{v}}_{r'}^{f,\psi(r')} \right)} \tag{11.10}$$

where γ denotes the digamma function. The updates are derived analogous to the derivation described in Section 7.4.

11.2 Experiment A: Learning the goal-directedness of Jazz harmony

The first Experiment shows that the goal-directedness of functional harmony is learnable from the observation of Jazz chord sequences. The headedness-induction model described in the previous section was used to induce 10 grammars from the observation of 300 Jazz chord sequences. As in the unsupervised grammar-learning experiments presented in Chapter 10, stochastic variational inference (Hoffman et al., 2013) was used with a batch size of 30 chord sequences. One epoch thus consists of 10 batches. The induction algorithm was run 10 times, because it is only guaranteed to converge to a local optimum. Each of the 10 trials observed the same chord sequences; the differences between the trials result from the randomness of stochastic variational inference.

The expected probability of right-headedness is shown in Figure 11.1 for each of the 10 trials and 100 batch updates (10 epochs). The convergence is slower than in the experiments presented in Chapter 10, but the inference procedure converges in each trial after at most 80 batch updates (8 epochs). Additional to start rules, duplication rules, and terminal rules, all 10 induced grammars use only right-headed rules instead of mixing right- and left-headed rules.

During learning, the model was free to choose any proportion of left- to right-headed rules. Therefore, the fact that all grammars induced by the headedness-induction model only use right headed rules shows that the goal-directedness of functional harmony is learnable from the observation of chord sequences without domain-specific prior knowledge. This can be seen as a data-driven confirmation of music-theoretical descriptions of functional harmony. The results also confirm the findings of the unsupervised grammar-learning experiments presented in Chapter 10. There, the either-headed grammars (that use both left- and right-headed rules) exhibit a slight tendency for right-headedness. Moreover, strictly right-headed grammars (that use only right-headed rules) perform better than either-headed grammars. More generally, the results suggest that goal-directedness is an example of an abstract principle

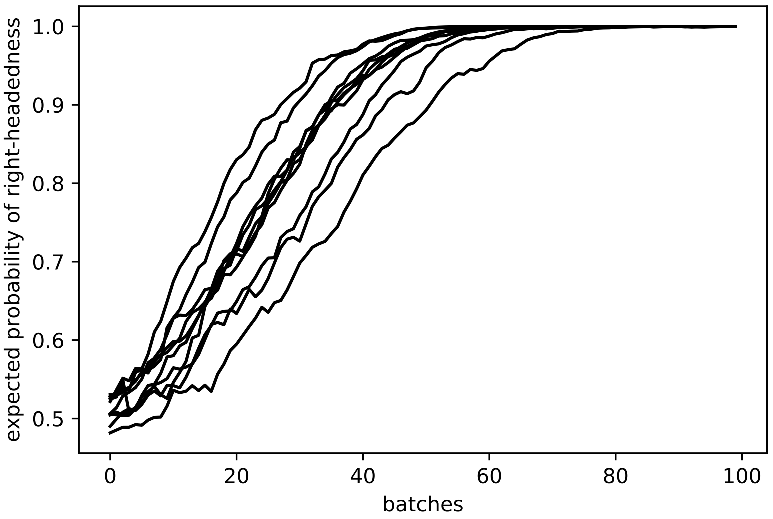


Figure 11.1 – Expected probability of right-headedness for each batch update during variational Bayesian grammar induction. Each line represents the headedness learning of one trial (10 trials in total).

that can be acquired by statistical learning. Therefore, it is plausible that Jazz musicians and listeners learned the goal-directedness of Jazz harmony from the interaction with Jazz standards.

11.3 Experiment B: Model verification using artificial data

To verify that the headedness-induction model is able to induce correct ratios of left-headed to right-headed rules, the model is applied to artificial terminal sequences generated from grammars with fixed ratios of left-headed to right-headed rules. The artificial grammars are similar to TPC-chord grammars; the structure of their rules is the same, and the parameterization of their rewrite probabilities is that of the headedness-induction model, but the artificial grammars use 30 symbols without internal structure as terminals and nonterminals instead of TPC-chords.

Each grammar is drawn from one of 15 grammar templates (i.e., distributions over grammars) that vary along two dimensions, 1) the Odds of Right-Headedness (ORH), and 2) the concentration of the Dirichlet prior over rule distributions $\zeta^{f,\rho}$. The values $1:1$, $2:1$, $3:1$, $5:1$, and $10:1$ are tested as ORH. For example, ORH of $10:1$ mean that right-headed rules are 10 times as likely as left-headed rules. The tested concentration parameters of the Dirichlet prior are 0.1, 1.0, and 10.0. The higher the concentration parameter is, the more different rules are applied to each nonterminal.

10 grammars are drawn for each template to generate 10 datasets of 300 terminal sequences. Each sequence consists of 20 artificial terminal symbols. Then, stochastic variational grammar inference is run for 5 epochs with a batch size of 30 sequences to recover the headedness of the grammar, using the headedness-induction model.

The results of the experiment are shown in Table 11.1. Because of the symmetric definition of the grammar model, the results are the same when left-headedness is switched with right-headedness. For the concentration parameter 0.1, the model is able to induce ORH close to the true ORH. The accuracy of the predictions increases and the variance of the predictions decreases when the ORH are closer to 1 : 1, that is when left-headed and right-headed rules are more balanced. The high accuracy verifies that the right-headedness induced in Experiment A reflects a property of Jazz chord sequences. The increasing uncertainty for increasingly unbalanced headedness might indicate why in Experiment A, the induced grammars do not use any left-headed rules, although a small percentage of left-headed rules would be more plausible: the headedness-inductions is less accurate when right-headed rules are used much more than left-headed rules.

For the concentration parameter 1.0, the model only infers the correct tendency of the correct headedness proportions. For a concentration of 10.0, it is not able to infer anything. These results suggest that grammar induction relies on low entropies of rule distributions. In other words, when the number of different rules that are applied to each nonterminal is low, then grammar induction is possible. If otherwise many different rules are applied to each nonterminal, then terminal sequences contain less information about the grammar they are generated from. This provides an explanation why rule distributions (for example those of the treebank shown in Figures 4.8 and 4.9) have low entropy. Otherwise, information about the grammar might not be encodable in the chord sequences and the grammar might not be learnable.

concentration	true ORH	predicted ORH	
		mean	std
0.1	1	0.98	0.08
0.1	2	2.03	0.13
0.1	3	3.02	0.21
0.1	5	4.94	0.54
0.1	10	8.56	0.87
1.0	1	0.98	0.11
1.0	2	1.18	0.10
1.0	3	1.32	0.17
1.0	5	1.24	0.16
1.0	10	1.25	0.16
10.0	1	0.97	0.10
10.0	2	1.05	0.12
10.0	3	1.07	0.12
10.0	5	1.05	0.10
10.0	10	0.98	0.05

Table 11.1 – Results for learning headedness proportions from artificial data. The first column shows the concentration parameter of the symmetric Dirichlet distributions that were used to sample the rule distributions $\boldsymbol{\zeta}^{r,\rho}$. The second column shows the Odds of Right-Headedness (ORH; as opposed to left-headedness) that were used to create the grammar from which the artificial terminal sequences were sampled. The third and the fourth columns show the mean and standard derivation of the odds of right-headedness learned from the artificial dataset. The results are analogous when right-headedness and left-headedness are swapped.

12 Contributions and conclusions

12.1 Original contributions

12.1.1 Theoretical contributions

This thesis presented and applied an integrated theory of music cognition, hierarchical music analysis, and artificial intelligence using the Bayesian interpretation of probability. The core feature of this theoretical framework is that it puts the individual in the center as an *ideal learner* or *ideal listener*. All probabilities calculated in this thesis are interpretable with respect to an ideal learner as the degrees of her rational belief. The theory emphasizes the role of musical background (as prior knowledge), and enables musicologists to quantitatively study subjective listening experiences of complex musical structure, with mathematical rigour.

Probabilistic Abstract Context-Free Grammar (PACFG) was presented as a flexible tool for cognitive modeling in general and grammar induction for Jazz harmony in particular. The formalization of grammar rules as partial rewrite functions lead to a concise description of the underlying mathematical theory in which derivation trees are represented as partial functions composed of rewrite rules. PACFG allows for a wide range of probabilistic models that are more powerful than conventional probabilistic context-free grammars. The utility of PACFG was demonstrated in the computational experiments in which it enabled, for instance, the usage of joint models of harmony and rhythm for grammar induction.

Parsing and inference for PACFG for was presented in the frameworks of semiring parsing and variational Bayesian inference. The thesis thereby bridges those theories by using semiring parsing to approximate expected values that are used for variational Bayesian inference. In particular, a semiring was defined which represents the distribution of all possible derivations of a sequence as a compact mathematical object. The combination of the theories as well as the semiring of derivation-tree distributions is expected to be useful for the implementation of prospective grammar models and the communication of inference procedures.

12.1.2 Empirical contributions

A dataset of hierarchical analyses of 150 complete Jazz standards, the Jazz Harmony Treebank (JHT), was created as a ground-truth database for model training and evaluation. Musical form was taken into account by proposing open constituents to create the analyses of the treebank. To ensure the interpretability of open constituents, an interface between a subset of musical form and functional harmony was defined and implemented. The JHT is an important step towards the development of an annotation standard for hierarchical structures of music. It is expected to be useful for future research and technology to train and evaluate grammar models, to scientifically study the details of Jazz harmony, and to provide a basis of examples useful for music education.

This study proposed two cognitively motivated probabilistic models that improved the state of the art for grammar models of harmony. The first model significantly improved the performance of harmony grammars by jointly modeling harmony and rhythm. Hence, it provides evidence that the interaction of several musical dimensions can significantly improve the acquisition of musical grammar. To implement such joint models, a product-grammar construction was proposed that is not tied to the specific musical dimensions of harmony and rhythm, but has the potential to integrate further dimensions such as meter, melody, and repetition structure as well.

The second model used a transpositionally invariant parameterization of rewrite probabilities to further improve the performance as well as the robustness of the grammar. The musical interpretation of that result is that a relative root representation and the possibility of modulation is beneficial for learning Jazz harmony. That in itself is not new for harmony theory, but it underpins the relevance of relative pitch relations for computational models and enables the quantification of their added value. In contrast to machine-learning approaches that augment data to obtain transpositionally invariant models (e.g, the training data is transposed to all 12 pitch classes), the approach presented in this thesis encodes transpositional invariance into the model's architecture. This leads to a cognitively more plausible model that is at the same time more efficient to train.

Two unsupervised grammar-learning experiments are reported in this thesis. The first experiment identified a model that is able to induce a grammar of Jazz harmony which performs nearly as well as the best grammar obtained via supervised learning (i.e., by observing the JHT). That model induces joint grammars of harmony and rhythm using a prior preference for simple rhythm and prior knowledge about the goal-directedness of functional harmony. Therefore, the first model shows that no style-specific knowledge is needed to induce a grammar for harmony from the observation of Jazz chord sequences. The second unsupervised experiment proposes a model for the induction of the directionality of harmonic dependencies. That model shows that the goal-directedness of Jazz harmony is learnable from the observation of Jazz chord sequences without domain-specific prior knowledge.

12.2 Directions for future research

One of the shortcomings is that the evaluation metrics tree accuracy and dependency accuracy are rough measure that do not reflect the ambiguity of music. The best grammar model achieves a tree accuracy of about 63% and a dependency accuracy of about 74%. From the viewpoint of natural language processing, these accuracy numbers might look like the model performs poorly, but its tree predictions are of high quality. Because of the high ambiguity of music, it is not plausible that any model which does not overfit the data would be able to achieve accuracies over 90%. The actual upper bound is, however, unknown and might be lower. Furthermore, the ground-truth data of the JHT was created by taking melody into account, and it is unknown how well a harmony grammar can predict a dependency structure without considering melody. It is therefore an interesting yet challenging topic for future research to jointly study the subjectivity of harmonic analyses as well as the interaction of harmony and melody together with the development of evaluation metrics. Current research studied related questions for chord labeling (Koops et al., 2019) that might be extendable to functional harmony. The study of the subjectivity of harmonic analyses is, however, very time-consuming and expensive, because multiple analyses of the same music have to created by different music experts.

A promising direction for future research is the explicit modeling of meter and harmonic upbeats in the rhythm grammar. The strength of the rhythm grammar as presented in this thesis is its simplicity; it is a simple yet powerful model that significantly improves the performance of harmonic grammars. The integration of meter and harmonic upbeats into this model has the potential to improve the performance further as the qualitative analyses of the tree predictions show. Grammar models of meter have already been implemented (e.g., McLeod and Steedman, 2017) and formal descriptions of integrated models are a topic of current research (Harasim et al., 2019b; Rohrmeier, 2020b).

For further improvements of the grammar models, the balancing and interaction of grammar components that individually model single musical dimensions can be optimized. In this study, the balancing of the grammar components was performed by hand, because an automatic balancing was not implementable with the current methodology. Future research can work on integrating more dimensions such as melody, key, and repetition structure and apply differential programming and stochastic gradient descent to train and balance the grammar components. Such approaches can then, for instance, incorporate artificial neural networks to learn complex parameterizations of probabilistic grammars (Kim et al., 2019).

A third idea for future research is the application of grammar induction to comparatively study Jazz and other styles such as Rock and Metal, Brazilian Choro, and Western Classical music. Particularly interesting would be the application of the headedness-induction model to Rock music, a style whose harmony is theorized to be not goal-directed. Further promising ideas for the application of grammar induction are discussed below.

12.3 Conclusion and prospects

This thesis has demonstrated that harmonic grammar can be induced from chord sequences without the need for music-specific predispositions. While music-specific predispositions are not required, joint consideration of harmony and rhythm was shown to be crucial to accurately acquire musical grammar. This underpins that harmony and rhythm are dependent; a consideration of rhythm reduces the number of plausible harmonic dependency structures. Similar dependencies are expected for other musical dimensions, for instance for melody which influences the harmonic function of a chord (e.g., whether a chord is a tonic). Indeed, music theorists have argued that considering musical dimensions in isolation as well as in interaction is essential to understand musical structure. The present thesis strongly supports this notion.

The goal-directedness of functional harmony was found to be beneficial for learning. Moreover, goal-directedness itself was learnable in the computational simulations. The importance of goal-directedness is (at least unconsciously) known to most improvising Jazz musicians. In my personal experience, good Jazz-improvisation teachers reinforce students to play towards a goal such as a tonic chord, instead of starting somewhere and playing away from it. For teaching improvisation to students, I made the observation that an advice to play towards a goal can help them a lot to transition from a rather meaningless playing of scales to an interesting story telling. It is astonishing to see such a close relation between practical music education and computational models of cognition; it affirms the importance of goal-directed thinking.

The computational models developed in this thesis offer a solid and resilient interface between music theory and empirical music research. Many of the modeling decisions taken in this thesis are based on music theory. Music-theoretical insights are operationalized and the "intuitive statistics" as well as the "folk psychology" of music theory are rigorously quantified (Cross, 1998; Neuwirth and Rohrmeier, 2016). Formal and computational modeling thus relies on music theory and, in converse, commonly confirms its statements as it is the case in this thesis. Furthermore, computational music theory has the advantage that all models are specified explicitly, transparently, and concisely in only a couple of paragraphs. This facilitates the scientific discussion about the model's implications as well as the usage of the model in computational applications. Moreover, various models can be compared to quantitatively investigate the importance of a musical property or relation. For instance, the importance of considering rhythm for learning harmonic grammar might be underestimated in qualitative studies but was corroborated in the present learning simulations. This thesis shows that it can be surprising how much musical structure is explainable via few simple assumptions and domain-general quantitative methodology such as provided by probability theory.

The computational cognitive models presented in this thesis complement traditional music theory in that they go beyond the question of how a piece might be heard to the question of how a way of hearing can be learned. Moreover, these models make the expressivity of music theory and music analysis available to study musicological questions quantitatively. Such questions could concern for instance comparative style analyses and historical developments. Computational cognitive models are therefore expected to play a major role in the rising field of musical corpus research.

Another interesting application of the grammar models developed in this thesis would be to apply the models in order to compare musical styles according to the learning difficulty of their respective grammars. The regularity of Jazz tunes facilitates an unsupervised learning of their structure and provides a grid-like fundament on which improvisation can be based. In contrast, most of Western classical music is composed and thus exhibits more complex and irregular structures than found in Jazz standards. Therefore, it would be plausible if grammar acquisition for Western classical music would benefit more from supervision during learning.

Two of the main criticisms of using context-free grammars for music analysis are that they might overstate the importance of recursion for musical structure and that one derivation tree is only able to represent a very limited set of relations between musical events. Clearly, generative models in general and context-free grammars in particular are modeling frameworks that make specific assumptions and are thus limited in their expressivity. However, probabilistic (abstract) context-free grammars constitute a local optimum in the spectrum between detailed music analysis and computational tractability. It is hard to represent dependencies over long time spans in simpler model classes such as linear models, and the computational complexity of more sophisticated model classes such as graph grammars is of a higher order of magnitude than context-free grammars. Future research can expand on the work presented in this thesis by taking additional systems of musical relations into account in order to further broaden the horizon of computational models of musical structure.

The question remains how a human mind benefits from making the effort to learn musical grammar. In natural language, grammatical structure is essential, because it is used to convey semantic meaning (Steedman, 2000). A child naturally acquires grammatical rules by communication with relatives and friends. The situation is less obvious for music: What is the advantage of understanding and interpreting hierarchical structure in music? Such an understanding facilitates, for example, the orientation in the music as well as a coordination between musicians. Knowledge about hierarchical phrase structure and musical form generally aids the orientation in the music. For example, it is easier to think about a chord as the last chord of the second A part in an AABA form than to think about it as a chord in the 15-th measure of a tune or as its 12-th chord. The AABA structure further implies that such a chord is likely to be a tonic chord, which is different for other structures such as ABAC. Orientation is of particular importance for improvisation, because an improvising musician cannot just memorize a tune note by note or chord by chord if she aims to embellish, reduce, or substitute the musical material. It is instead plausible that musicians use an abstract representation of a tune for improvisation which enables them to freely explore the musical space by keeping the connection to other musicians at the same time. Musical grammar describes the construction of such abstract representations. The goal-directedness of functional harmony further helps to coordinate collective improvisation, because how musicians prepare a chord is not as important as that they prepare the same chord when playing together. The same holds for collectively achieving closure. Hence, the regular formal structure of Jazz standards and the goal-directedness of functional harmony can be explained by their facilitation of individual and collective improvisation. Moreover, improvising dancers use musical grammar similarly to improvising musicians in order to anticipate the music and plan their moves.

The plausibility that musical grammar is learned instead of being innate is further supported by the present finding that it is relatively easy to acquire musical grammar through statistical learning. The experiments showed that a grammar of high quality was inducible from a small sample of only 300 Jazz standards. Moreover, the quick convergence of the inference procedure suggests that the sample can be further reduced without compromising the quality of the grammar. The findings of this thesis therefore indicate that the poverty of the stimulus argument is not applicable to music. However, direct comparisons between language and music (e.g., Katz and Pesetsky, 2011) need to be treated with caution since the rules of musical grammar appear to be more subtle and less strict than those of language.

One explanation why musical grammars might be easier to learn than grammars for natural language is the massively reduced lexicon. The PC-chord grammar used a vocabulary of 144 chord symbols. In contrast, native-speakers of English use a vocabulary of more than 10,000 words. The role of chord symbols might thus be more similar to the role of parts of speech than to words. In that understanding, chord symbols categorize chords; no additional categorization might be required to learn musical grammar. This can be studied in the future by the application of formal grammar models on the note level.

Bayesian models of cognition provide a powerful coherent toolbox for interdisciplinary research on music. Their theoretical strengths are to favor interpretability and explicit representation of symbolic knowledge over ad-hoc black-box modeling. I expect future research to further push the boundaries of which musical structures can be characterized by computational models. The study of music is a great opportunity to simultaneously explore the details of musical structure, technology for artificial intelligence, and concepts applicable to music education, and to gain insights into the human mind.

A Appendix: related published articles

A GENERALIZED PARSING FRAMEWORK FOR GENERATIVE MODELS OF HARMONIC SYNTAX

Daniel Harasim[1,2] **Martin Rohrmeier**[1,2] **Timothy J. O'Donnell**[3]
[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne, Switzerland
[2] Institut für Kunst- und Musikwissenschaft, TU Dresden, Germany
[3] Department of Linguistics, McGill University, Canada
daniel.harasim@epfl.ch

ABSTRACT

Modeling the structure of musical pieces constitutes a central research problem for music information retrieval, music generation, and musicology. At the present, models of harmonic syntax face challenges on the tasks of detecting local and higher-level modulations (most previous models assume a priori knowledge of key), computing connected parse trees for long sequences, and parsing sequences that do not end with tonic chords, but in turnarounds. This paper addresses those problems by proposing a new generative formalism Probabilistic Abstract Context-Free Grammars (PACFGs) to address these issues, and presents variants of standard parsing algorithms that efficiently enumerate all possible parses of long chord sequences and to estimate their probabilities. PACFGs specifically allow for structured non-terminal symbols in rich and highly flexible feature spaces. The inference procedure moreover takes advantage of these abstractions by sharing probability mass between grammar rules over joint features. The paper presents a model of the harmonic syntax of Jazz using this formalism together with stochastic variational inference to learn the probabilistic parameters of a grammar from a corpus of Jazz-standards. The PACFG model outperforms the standard context-free approach while reducing the number of free parameters and performing key finding on the fly.

1. INTRODUCTION

The modeling of non-local relations between musical objects such as notes and chords constitutes a central research problem for music information retrieval, music generation, and music analysis. Hierarchical models express these relations by assuming a latent hierarchical structure [19,22–24,30,31]. Consider for example the Jazz chord sequence Am^7 D^7 G^7 C$^\triangle$ where C$^\triangle$ denotes a major-seventh chord. Since the first three chords form a II V I sequence with reference to G^7 which is the dominant in C major, they form a *dominant phrase* [24]. The dominant phrase as a whole then refers to the tonic chord C$^\triangle$. All four chords together thus form a *tonic phrase*.

Figure 1 presents a syntactic analysis of the A-part of the Jazz-standard *Afternoon in Paris* following the approach from [22]. It illustrates the idea of how pieces can be decomposed into hierarchically-structured *constituents* which stand in part-whole relationship with one another. Subdominant, dominant, and tonic phrases are denoted by the scale degrees II, V, and I, respectively. Note that the subsequence Cm^7 F^7 B♭$^\triangle$ is both a tonic progression in B♭ major and a dominant progression in E♭ major. It forms a dominant phrase in A♭ major together with B♭m^7 and E♭^{7}.

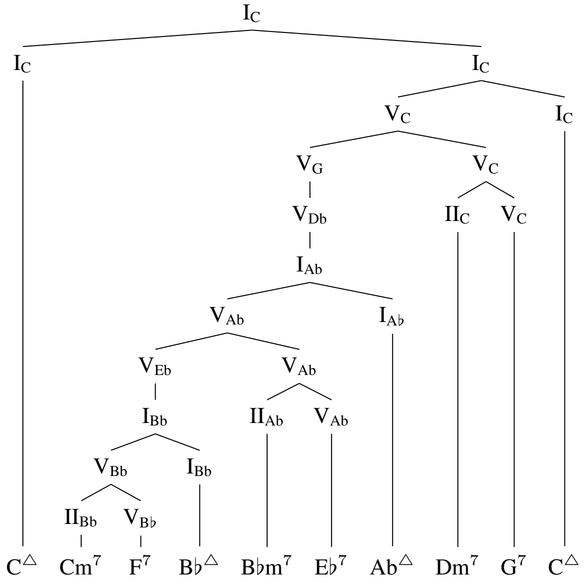


Figure 1. Hierarchical analysis of the A-part of the Jazz-standard *Afternoon in Paris*.

Models of harmonic syntax similar to Figure 1 have been successfully applied to melody harmonization [16], chord inference from audio [5, 6], and harmonic similarity [7]. There is also some empirical evidence for the psychological reality of hierarchical structures in music [15, 25]. While earlier theoretical and psychological work on hierarchical models has provided important insight about musical structure, computational implementation of these models to date has been limited to relatively small datasets. Earlier work includes applications to monophonic melodic data [21], a corpus of 39 blues chord progressions with a maximum of 24 chords per progression [12], or a dataset of 76 chord progressions (avg. length 40) from Jazz-standards that was restricted to subsequences of pieces that did not change key [4]. All these earlier approaches assume the knowledge of the key of the pieces a priori.

In computational linguistics, Context-Free Grammars (CFGs) are a standard way of modeling hierarchical constituent structure. They formalize constituent structures using *rewrite rules* denoted by long right arrows. The rule $X \longrightarrow Y\ Z$ for example states that the constituent X *consists of* the two constituents Y and Z. The existence of natural language treebanks makes it possible to read off the grammatical rewrite rules including their frequencies from syntactical analyses by experts. At present, there are music databases of simplified Schenkerian analyses [13], syntactic analyses of melodies based on the generative theory of tonal music [8], and annotated harmonic functions [4]. However, to the best of our knowledge there is currently no dataset of hierarchically analyzed chord sequences by human experts that could serve for the training or the evaluation of models of harmonic syntax. As a consequence, there exist no comparisons of models of harmonic syntax against expert analyses.

In the following, we introduce Abstract Context-Free Grammars (ACFGs), a generalization of the CFG framework designed to account for feature structures characteristic of musical categories. A first model of Jazz harmony is proposed in this framework that covers full pieces by incorporating modulations (i.e., changes in key). We train the model in a semi-supervised fashion on a dataset of Jazz-standards and evaluate it on a small set of hand-annotated hierarchical analyses. We further propose a solution for handling sequences that do not end with tonic chords, but in turnarounds. Simulations demonstrate that the ACFG model is able to outperform a PCFG model of the dataset. The implementation of the algorithms developed in this study are publicly available as a package of the Julia programming language [1]. [1]

2. OVERVIEW OF THE APPROACH

While the CFG framework has proven invaluable in computational linguistics, categories and part-whole relations between musical constituents have properties not possessed by linguistic structures. Musical categories such as scale degrees, for example, are equipped with an arithmetic structure that corresponds to musical transposition.

In the following, we refer to context-free rules of the form $X \longrightarrow Y\ X$ as a *preparation* of X by Y. The preparation of the scale degree $\mathrm{V}_{\mathrm{B}\flat}$ by $\mathrm{II}_{\mathrm{B}\flat}$ in *Afternoon in Paris* (see Figure 1) for example is a concrete realization of the general principle that any category x_k consisting of a scale degree x and a key k can be prepared by an ascending diatonic fifth $(x+4 \bmod 7)_k$. [24]. In addition to facts such as these, a framework for modeling musical structure has to account for the fact that the musical categories and rewrite rules are grouped into key-independent classes. For example, both $V_{\mathrm{B}\flat}$ and $V_{\mathrm{A}\flat}$ are fifth scale degrees. The probabilities of the application a rule to $V_{\mathrm{B}\flat}$ and $V_{\mathrm{A}\flat}$ should therefore be related.

This paper introduces Abstract Context-free Grammars (ACFGs), a modeling framework with a greater flexibility than CFGs. In particular, in ACFGs constituent categories are allowed to be of any data type and the rules are generalized partial functions. Unlike standard context-free rules, ACFG rules can therefore take advantage of the algebraic structure of categories. Probabilistic ACFGs extend probabilistic CFGs with the ability to express a wider range of probability distributions over rules.

3. ABSTRACT CONTEXT-FREE GRAMMARS

3.1 Definitions

Definition 1. A *(non-probabilistic) Abstract Context-free Grammar* (ACFG) $G = (T, C, C_0, \Gamma)$ consists of a set T of *terminal symbols*, a set C of *constituent categories*, a set of *start categories* $C_0 \subseteq C$, and a set of partial functions

$$\Gamma := \{\, r \mid r : C \nrightarrow (T \cup C)^* \,\},$$

called *rewrite rules* or *rewrite functions*. The arrow $\nrightarrow$ is used throughout the paper to denote partial functions. A sequence $\beta \in (T \cup C)^*$ can be *generated in one step* from a sequence $\alpha \in (T \cup C)^*$ by the application of a rewrite function $r \in \Gamma$, denoted by $\alpha \longrightarrow_r \beta$, if there exist $\alpha_1, \alpha_2 \in (T \cup C)^*$ and $A \in C$ such that $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 r(A) \alpha_2$. A sequence of rewrite rules $r_1 \dots r_n$ is called a *derivation* of a sequence of terminals $\alpha \in T^*$ if there exists a start category $\alpha_1 \in C_0$, and $\alpha_2, \dots, \alpha_n \in (C \cup T)^*$ such that

$$\alpha_1 \longrightarrow_{r_1} \alpha_2 \longrightarrow_{r_2} \cdots \longrightarrow_{r_n} \alpha,$$

where r_i is always applied to the leftmost category of α_i for $i \in \{1, \dots, n-1\}$. The set of derivations of α is denoted by $D(\alpha)$. The language of the grammar G is the set of terminal sequences that have a derivation in G.

Note that if C is finite, the languages that can be described by ACFGs are exactly the languages that can be described by standard context-free grammars (CFGs). For each ACFG with finite C, a CFG with rule set R can be constructed by dividing each rewrite function with domain cardinality k into k standard context-free rewrite rules,

$$R := \bigcup_{r \in \Gamma} \{\, (A, \alpha) \in C \times (T \cup C)^* \mid r(A) = \alpha \,\}.$$

Definition 2. A *Probabilistic Abstract Context-free Grammar* (PACFG) is an ACFG where each category $A \in C$ is associated with a random variable X_A over rewrite functions r such that $\mathbb{P}(X_A = r)$ is positive if and only if $r(A)$

[1] https://github.com/dharasim/GeneralizedChartParsing.jl

is defined, that is A is in the domain of r, $A \in \mathrm{dom}(A)$. The probability $p(d)$ of a derivation $d = r_1 \dots r_n$ of a sequence of terminal symbols $\alpha \in T^*$ is defined as the product $\prod_{i=1}^{n} \mathbb{P}(X_{A_i} = r_i)$ where in each step r_i is applied to a category $A_i \in C$. The probability of α is then defined as $p(\alpha) = \sum_{d \in D(\alpha)} p(d)$.

Note that PACFG categories can share the same probability distribution over rewrite functions without rewriting to exactly the same right-hand sites. This important property allows us to model the structural relations between musical keys. We use this property in Section 4 to build a model that abstract chords sequences from their concrete scale by defining the probability that a rewrite function is applied to a scale degree independently of its key. The sharing of probability mass between rules additionally reduces the number of free parameters of a PACFG model.

To illustrate the different learning capabilities of PCFG and PACFG models, consider a toy PCFG with nonterminal symbols $C = \{ S, A, B \}$, start symbol S, terminal symbols $T = \{ a, b \}$, and rules $S \longrightarrow A \mid B$, $A \longrightarrow A\ A \mid a$, and $B \longrightarrow B\ B \mid b$. The grammar thus generates sequences that solely consist either of as or bs. In a classical PCFG setting, no probability mass is shared between rules, but each rule has its separate probability. However, in the process of inferring the probabilities of the rules from data, it might be desirable to generalize the rules $A \longrightarrow A\ A$ and $B \longrightarrow B\ B$ to a meta rule $x \longrightarrow x\ x$ where $x \in \{ A, B \}$ and to put probability mass on this abstract entity. In that way, the grammar can learn something about $A \longrightarrow A\ A$ when it observes $B \longrightarrow B\ B$ and vice versa. The PACFG version of the PCFG presented above addresses the problem by replacing the classical context-free rules by the partial functions r_1, r_2, r_3, r_4, and r_5 with $r_1(S) = A$, $r_2(S) = B$, $r_3(x) = x\ x$ for $x \in \{ A, B \}$, $r_4(A) = a$, and $r_5(B) = b$. Analogously, a PACFG of Jazz chord sequences can generalize classical rewrite rules so that their probabilities do not depend on the keys of their left-hand sides to model transpositional invariance.

3.2 Parsing

Parsing a sequence of terminal symbols with respect to a formal grammar is the task of computing the distribution of parse trees conditioned on this sequence. Many parsers are based on versions of the CYK algorithm that assumes grammars to be given in Chomsky normal form. Since grammar transformations into Chomsky normal form considerably blow up the grammar, the here presented parser transforms grammars on the fly during parsing, similar to the transformation presented in [18]. Each rule of the form $A \longrightarrow B_1 \dots B_k$ is transformed into a set of *states* $s_i = B_1 \dots B_i$ for $1 \leq i \leq k$, a transition function

$$\mathrm{tran} : S \times (T \cup C) \to S, \quad \mathrm{tran}(s_i, B_{i+1}) = s_{i+1}$$

and a completion function $\mathrm{comp} : S \to 2^C$ such that $\{ A \} \subseteq \mathrm{comp}(s_k)$, where S denotes the set of all states. Note that the states and the transition function form a search trie where the completion function checks if there is a rewrite rule that has a sequence of terminal symbols and categories as its right-hand side. This trie data structure leads to a compact representation of the forest of all trees for a given input sequence. More generally, the parser can handle any transition and completion functions derived from finite-state automata, see [14].

$$\begin{array}{lll}
\text{items:} & \text{edges} & [s,i,j] \quad \text{for } s \in S \\
 & \text{constituents} & [A,i,j] \quad \text{for } A \in C \\
 & & \text{for and } i,\, j \in \{1, \dots, |\alpha|+1\}
\end{array}$$

$$\begin{array}{lll}
\text{goal items:} & [A, 1, |\alpha|+1] & \text{for } A \in S \\[1em]
\text{axioms:} & \dfrac{}{[\alpha_i, i, i+1]} & \text{for } i \in \{1, \dots, |\alpha|\} \\[1em]
\text{introduce edge:} & \dfrac{[A,i,j]}{[s,i,j]} & s = \mathrm{tran}(s_0, A) \\[1em]
\text{complete edge:} & \dfrac{[s,i,j]}{[A,i,j]} & A \in \mathrm{comp}(s) \\[1em]
\text{fundamental rule:} & \dfrac{[s,i,j] \quad [A,j,k]}{[s',i,k]} & \mathrm{tran}(s,A) = s'
\end{array}$$

Figure 2. Description of the parsing algorithm in the parsing as deduction framework. Existing Constituents can start the parser to read a sequence of terminal symbols and categories by the *introduce edge* rule. The *fundamental rule* is then recursively applied to extend these sequences. The *complete edge* rule eventually merges sequences to single constituents if they are the right-hand side of a grammar rule.

In the following, a generic bottom-up parsing algorithm for abstract grammars is presented in the parsing as deduction framework using the above defined transition and completion functions [3, 29]. The parsing as deduction framework is a meta-formalism to state and compare different parsing algorithms. It views the parses of a sequence as logical deductions of goals from axioms by using constituents as atomic logical formulas. The formula $[\mathrm{I}_{\mathrm{B}\flat}, 2, 5]$ for example states the existence of a constituent with category $\mathrm{I}_{\mathrm{B}\flat}$ that spans over the second, third, and fourth terminal symbol. This formula is true in the analysis presented in Figure 1 because that analysis contains a constituent with label $\mathrm{I}_{\mathrm{B}\flat}$ over the span from the second to the forth leaf chord. The goals are constituents that span the full sequence and come from the set of start categories. The axioms are formulas of the form $[t_i, i, i+1]$ for each terminal in the input sequence $t_1 \dots t_n$. The parsing strategy such as bottom-up parsing or Earley parsing is encoded in the deduction rules. These rules are denoted by a set of atomic formulas over a horizontal line, an atomic formula under this line, and an optional side condition (see Figure

2). The formula under the line can be deduced from the formulas above if the side condition holds.

The proposed algorithm makes use of two different kinds of atomic formulas: edges (not yet completed constituents) and constituents. A state $s \in S$ together with a start index i and an end index j is called an *edge* and denoted by $[s, i, j]$. Analogously, a category $A \in C$ together with start and end indices i and j is called a *constituent* and denoted by $[A, i, j]$. Figure 2 shows the axioms, goal items, and the deduction rules of our algorithm.

3.3 Inference of Rule Probabilities

In this section, we give an overview of an inference algorithm for the rule probabilities $\mathbb{P}(X_A = r)$. Let $\Gamma_A = \{\, r \in \Gamma \mid A \in \operatorname{dom}(r) \,\}$ be the set of rewrite functions whose domain contains the constituent category A. We place a Dirichlet distribution on the probability vector describing the distribution over Γ_A, $\vec{\theta}_{\Gamma_A} \sim \text{Dirichlet}(\vec{\alpha}_{\Gamma_A})$ for pseudocount vector $\vec{\alpha}_{\Gamma_A}$. The inference problem is to compute the posterior distribution over this set of probability vectors, given the data D and pseudocounts $\{\vec{\alpha}_{\Gamma_A}\}$,

$$p(\{\vec{\theta}_{\Gamma_A}\} \mid D, \{\vec{\alpha}_{\Gamma_A}\}) \propto p(D \mid \{\vec{\theta}_{\Gamma_A}\}) p(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\alpha}_{\Gamma_A}\}),$$

where $\{\vec{\theta_{\Gamma_A}}\}$ is an abbreviation for $\{\vec{\theta_{\Gamma_A}}\}_{A \in C}$, etc. *Variational Bayesian inference* (VB) is used to approximate this posterior distribution [2, 11, 32]. We introduce an approximating *variational distribution* $q(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\nu}_{\Gamma_A}\})$ with *variational parameters* $\{\vec{\nu}_{\Gamma_A}\}$ over our target hidden variables (rule weights) and minimize the Kullback-Leibler divergence between this approximation and the true posterior,

$$D_{\mathrm{KL}}(q(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\nu}_{\Gamma_A}\}) \mid\mid p(\{\vec{\theta}_{\Gamma_A}\} \mid D, \{\vec{\alpha}_{\Gamma_A}\})),$$

by adjusting the variational parameters $\{\vec{\nu}_{\Gamma_A}\}$.

Following [17], we approximate the distribution over each probability vector with a Dirichlet distribution $\vec{\theta}_{\Gamma_A} \mid \vec{\nu}_{\Gamma_A} \sim \text{Dirichlet}(\vec{\nu}_{\Gamma_A})$, and make use of the *mean-field approximation*

$$q(\{\vec{\theta}_{\Gamma_A}\} \mid \{\vec{\nu}_{\Gamma_A}\}) = \prod_{A \in C} p(\vec{\theta}_{\Gamma_A} \mid \vec{\nu}_{\Gamma_A}).$$

We minimize the Kullback-Leibler divergence with a coordinate descent algorithm similar to the expectation-maximization algorithm. First, we compute the expectation of the counts of rule usages in the data under our current setting of the variational parameters, $\mathbb{E}_q[\#(r, D)]$ where $\#(r, D)$ is the number of times that rule r was used to generate the data D, and then we update our variational parameters based on these expectations. Since all of our distributions are in the exponential family, it can be shown that the optimal update is given by the equation $\hat{\vec{\nu}}_{\Gamma_A} = \vec{\alpha}_{\Gamma_A} + \mathbb{E}_q[\#(r, D)]$ [2]. In other words, we set the pseudocounts of our variational distributions equal to the expected number of rule usages plus the pseudocount for each rule in the prior distribution.

Under the standard coordinate-ascent algorithm given in [17], expected counts must be computed for the whole corpus before updating using the equation above. Hoffman et al. [9] propose a stochastic variant of the standard variational (inspired by *stochastic gradient descent*) where updates are computed with respect to randomly sampled *minibatches* of the data. We make use of this *stochastic variational Bayes* algorithm in the results reported below.

4. A GENERATIVE MODEL OF JAZZ HARMONY

This section presents a PACFG $G = (T, C, C_0, \Gamma)$ that models the syntax of Jazz harmony following the proposal in [24]. That work addressed the problem of finding a restrictive grammar that describes the full variety of syntactic relations in the musical idiom of Jazz-standards. The set of terminal symbols T is a set of pairs describing chords each of which consists of the root of the chord and a string describing the chord form—one of: a major triad, a major-seventh chord, a major sixth chord, a dominant-seventh chord, a minor triad, a minor-seventh chord, a half-diminished-seventh chord, a diminished seventh-chord, an augmented triad, or a suspended chord.

In the following, $\mathbb{Z}_n$ denotes the ring of integers modulo $n \in \mathbb{N}$. The categories are modeled as pairs of scale degrees and keys, $C = \mathbb{Z}_7 \times K$, where a key consists of a pitch class representing its root and a string describing its mode, $K = \mathbb{Z}_{12} \times \{\, \text{major}, \text{min} \,\}$. Scale degrees are denoted by roman numerals from I to VII. All categories with scale degree I are start symbols, $C_0 = \{\, \text{I} \,\} \times K$. Let $k \in K$ denote an arbitrary key. The set of rewrite functions Γ consists of *prolongation*,

$$\textsc{prolong}(\langle x, k\rangle) = \langle x, k\rangle\ \langle x, k\rangle$$

for $x \in \mathbb{Z}_7$, *diatonic preparation*,

$$\textsc{diat-prep}(\langle x, k\rangle) = \langle x + 4 \bmod 7, k\rangle\ \langle x, k\rangle$$

for $x \in \mathbb{Z}_7 \setminus \{\, \text{IV} \,\}$, *dominant preparation*,

$$\textsc{dom-prep}(\langle x, k\rangle) = \langle \text{V}, \mu(x, k)\rangle\ \langle x, k\rangle$$

for $x \in \mathbb{Z}_7 \setminus \{\, \text{I} \,\}$ where $\mu(x, k)$ denotes the modulation from k into the key of scale degree x (e.g. $\mu(\text{II}, (0, \text{maj})) = (2, \text{min})$, the key of the second scale degree of C major is D minor), *plagal preparation*,

$$\textsc{plagal-prep}(\langle \text{I}, k\rangle) = \langle \text{IV}, k\rangle\ \langle \text{I}, k\rangle,$$

modulation,

$$\textsc{modulation}(\langle x, k\rangle) = \langle \text{I}, \mu(x, k)\rangle,$$

mode change,

$$\textsc{mode-change}(\langle \text{I}, (r, m)\rangle) = \begin{cases} \langle \text{I}, (r, \text{min})\rangle, & \text{if } m = \text{maj} \\ \langle \text{I}, (r, \text{maj})\rangle, & \text{if } m = \text{min}, \end{cases}$$

for $r \in \mathbb{Z}_{12}, m \in \{\, \text{maj}, \text{min} \,\}$, *diatonic substitution*,

$$\textsc{diat-subst}(\langle x, (r, m)\rangle) = \begin{cases} \langle \text{VI}, (r, m)\rangle, & \text{if } x = \text{I}, m = \text{maj} \\ \langle \text{III}, (r, m)\rangle, & \text{if } x = \text{I}, m = \text{min} \\ \langle \text{IV}, (r, m)\rangle, & \text{if } x = \text{II} \\ \langle \text{VII}, (r, m)\rangle, & \text{if } x = \text{V} \end{cases}$$

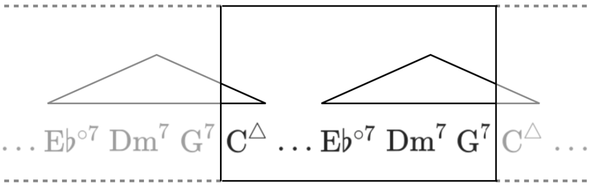


Figure 3. Parsing the turnaround of *All of me*

for $x \in \{\,\mathrm{I}, \mathrm{II}, \mathrm{V}\,\}$, $r \in \mathbb{Z}_{12}$, $m \in \{\,\mathrm{maj}, \mathrm{min}\,\}$, and *dominant substitution*,

$$\textsc{dom-subst}_i(\langle \mathrm{V}, (r, m)\rangle) = \langle \mathrm{V}, (r + i \bmod 12, m)\rangle$$

for $r \in \mathbb{Z}_{12}$, $m \in \{\,\mathrm{maj}, \mathrm{min}\,\}$, and $i \in \{\,3, 6, 9\,\}$. Additionally, Γ contains appropriate termination rules $C \twoheadrightarrow T$ according to standard Jazz harmony theory (e.g. seventh-chord-termination$(\langle 4, (0, \mathrm{maj})\rangle) = \mathrm{G}^7$, see [20] for further explanation). The distribution of $X_{\langle x,k\rangle}$ over rules rewriting the category $\langle x, k\rangle$ is defined as a categorical distribution such that $\mathbb{P}(X_{\langle x,k\rangle} = r) = \mathbb{P}(X_{\langle x,k'\rangle} = r)$ for all scale degrees x, rules r, and keys k, k' that have the same mode. That is, the probability of r rewriting $\langle x, k\rangle$ does not depend on the root of k which enables the model to learn the parameters of its probability distributions key-independently.

These grammar rules can be grouped into three classes: the prolongation rule, preparation rules, and substitution rules. Preparation rules create categories that for the listener generate the expectation to hear the prepared chord. Substitution rules substitute chords for other chords that fulfill an equivalent function inside the sequence such as tritone substitutions of dominants in Jazz.

5. THE TURNAROUND PROBLEM

A lead-sheet of a Jazz-standard consists of a melody together with a chord sequence describing the fundamental harmonic structure of the piece. The chord sequence is repeated multiple times in a performance. While some lead-sheets end with tonic chords, others include harmonic upbeats to the first chord of the piece at the end of the sheet, called *turnarounds*. The final chord of a performances is nevertheless usually a tonic chord. The lead-sheet of the Jazz-standard *All of me* starts for example with a $\mathrm{C}^{\triangle}$ chord and ends with the turnaround $\mathrm{E}\flat^{\circ 7}$ Dm^7 G^7.

The grammar of Jazz harmony proposed above assumes that pieces end with a tonic chord. Therefore, a simple implementation of this grammar would not able to parse lead-sheets that end in turnarounds. We solve this problem by *cyclic parsing*, meaning that we assume that constituents can have spans from the end of a piece back to the beginning, see Figure 3.

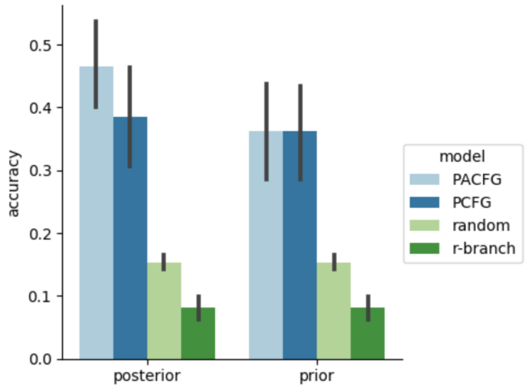


Figure 4. Tree accuracy plot

6. EXPERIMENTS

6.1 Dataset

The model is evaluated using the *iRealPro* dataset of Jazz-standards. [2] This dataset consists of 1173 chord sequences electronically-encoded by the Jazz musician community including metadata such as the titles, composers, and keys. The sequences were collected and converted into the Humdrum format [10] by Daniel Shanahan and Yuri Broze [28], and are available online. [3] For other research that uses this dataset see [26, 27]. The chord forms in the *iRealPro* dataset include information about nineths and elevenths that are not considered in this study.

The subset of 394 Jazz-standards that consist of at most 40 chords was considered to train the models. 34.52% (136) of these pieces were parsable using the standard approach and 90.61% (357) pieces were parsable using the cyclic parsing approach described above. Less then 55% of the considered Jazz-standards therefore end in turnarounds.

6.2 Tree Accuracy Evaluation

We compare four models: (i) the proposed PACFG model that uses a representation of rules independent of key, (ii) its PCFG counterpart the rules of which are not independent of key, (iii) a baseline of randomly generated trees, and (iv) a *right-branching baseline* in which all constituents split into a constituent on the left and a terminal symbol on the right.

The models are trained on the 357 cyclic parsable sequences using minibatches of 8 sequences. They are evaluated on 13 pieces hand-annotated by the authors. We report the predicted tree accuracy. That is the precision of correctly predicted spans of internal tree nodes. A span of a tree node is defined as the start index of its leftmost leaf together with the end index of its rightmost leaf.

Figure 4 shows the means of the tree accuracies including 95% confidence intervals as error bars. The right-branching baseline performs at an accuracy level under 10%. The random baseline performs slightly better at an

[2] https://irealpro.com

[3] https://musiccog.ohio-state.edu/home/index.php/iRb_Jazz_Corpus

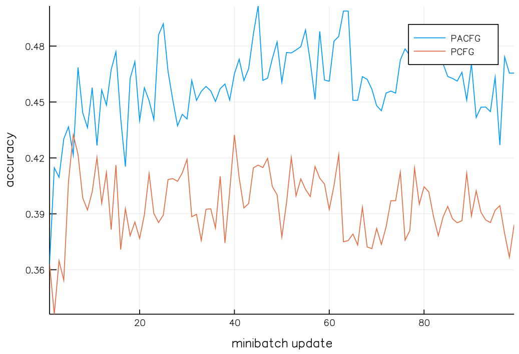


Figure 5. Predicted tree accuracy for each minibatch update. Note that the y-axis displays only values between 33% and 50%.

accuracy level of 15.35% Under a uniform prior, both the PACFG and the PCFG model perform at an accuracy level of 36.30% a priori of the data. As opposed to the trained PCFG model that only improves its performance by about 3% (in comparison to the uniform prior) reaching an accuracy of 39.43%, the trained PACFG model improves by about 10% (in comparison to the uniform prior) reaching an accuracy of 45.95%. The PACFG model was thus able to learn more from the data than the PCFG model. Note that since the PCFG model does not abstract the grammar rules from the concrete key wherein they are applied, the number of free parameters of the PCFG model is approximately 12 times higher than the number of free parameters of the APCFG model.

Despite the fact that the PACFG model learns key-independently, it is still much simpler than models that produce state-of-the-art parsing results in computational linguistics. In particular, state-of-the-art models in computational linguistics typically make use of conditioning information beyond the parent constituent categories used in the PACFG model—such as larger tree fragments, conditioning on heads and/or adjacent elements in the string, state-splitting, and other richer contextual information. We anticipate that the inclusion of similar structures into musical parsing models will lead to similar improvements in performance.

Figure 5 shows the mean predicted tree accuracies of the PACFG and the PCFG models for each minibatch update. Note that this figure is produced using a stochastic algorithm and is therefore inherently noisy. We see that the stochasticity of the inference algorithm leads to random jumps of the accuracy up to 0.5%. The models appear to do most of their learning in the first 10 minibatches.

6.3 Performance Diagnosis using Scale Degree Frequencies

Figure 6 shows the expected frequency of scale-degree use in the whole corpus. The scale degrees VI in major and III in minor are more frequently used by the model than expected. Because these scale degrees are substitutions for the first scale degrees and because they enable modulations into the relative key (e.g. from C major to A minor and vice versa), the model may be using them to alternate between relative keys. The prominence of the VII in minor keys is probably related to the fact that it has a dominant-seventh chord form. The model may be interpreting a I in major as a III in the relative minor key that is then prepared by the VII in minor. For example, the simple chord transition G^7 $C^{\triangle}$ would in this case be derived by

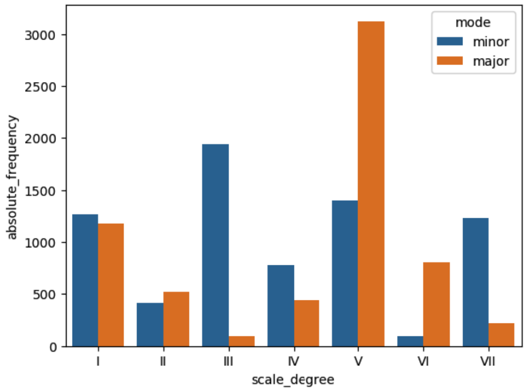


Figure 6. Expected usage of scale degrees to parse the full training dataset

$$\mathrm{I}_a \longrightarrow \mathrm{III}_a \longrightarrow \mathrm{VII}_a\ \mathrm{III}_a \longrightarrow \mathrm{G}^7\ \mathrm{III}_a \longrightarrow \mathrm{G}^7\ \mathrm{C}^{\triangle}.$$

7. CONCLUSION AND FUTURE RESEARCH

The research presented here introduced a new general grammar and parsing framework tailored to the needs of music and showed how to perform inference for such a model.

Experiments show that in contrast to standard context-free models, the proposed model is able to learn characteristic structures of the observed data. To the best of our knowledge, this is the first computational approach that automatically performs hierarchical analyses of chord sequences and evaluates them on analyses by human experts.

This paper lays the groundwork for more advanced models of harmonic syntax. Our future research will in particular focus on expanding the dataset of hand-annotated expert analyses to provide significance tests of the performance comparison of different models, for example. Further studies can use the tools developed here to build models of unsupervised grammar induction, joint models of multiple musical levels of musical structure like harmony and rhythm, and models of musical structure that have more complex dependencies than those representable in simple tree structures.

8. ACKNOWLEDGEMENTS

Financial support for the research presented in this article has in part been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the *Zukunftskonzept* at TU Dresden funded by the *Exzelleninitiative* of the *Deutsche Forschungsgemeinschaft*.

9. REFERENCES

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM review*, 59(1):65–98, 2017.

[2] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational Inference: A Review for Statisticians. *arXiv*, (arXiv:1601.00670), 2017.

[3] Joshua T Goodman. *Parsing inside-out*. PhD thesis, 1998.

[4] Mark Granroth-Wilding and Mark Steedman. A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43(4):355–374, 10 2014.

[5] W Bas De Haas. *Music information retrieval based on tonal harmony*. PhD thesis, 2012.

[6] W Bas De Haas, Jos Pedro Magalhães, and Frans Wiering. Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. *International Society for Music Information Retrieval Conference (ISMIR)*, (Ismir):295–300, 2012.

[7] W Bas De Haas, Martin Rohrmeier, and Frans Wiering. Modeling Harmonic Similarity using a Generative Grammar of Tonal Harmony. *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, 2009.

[8] Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. Musical Structural Analysis Database Based on Gttm. In *Proceedings of the 15th Conference of the International Society for Music Information Retrieval*, pages 325–330, 2014.

[9] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[10] David Brian Huron. *The Humdrum Toolkit: Reference Manual*. Center for Computer Assisted Research in the Humanities, 1994.

[11] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakola, and Lawrence K Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37:183–233, 1999.

[12] Jonah Katz. Harmonic Syntax of the Twelve-Bar Blues Form. *Music Perception: An Interdisciplinary Journal*, 35(2):165–192, 2017.

[13] Phillip B Kirlin and David D Jensen. Using Supervised Learning to Uncover Deep Musical Structure. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1770–1776, 2015.

[14] Dan Klein and Christopher D. Manning. Parsing and Hypergraphs. *Proceedings of the 7th International Workshop on Parsing Technologies (IWPT-2001)*, (c):351372, 2001.

[15] Stefan Koelsch, Martin Rohrmeier, Renzo Torrecuso, and Sebastian Jentschke. Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, 110(38):15443–15448, 2013.

[16] Hendrik Vincent Koops, Jos Pedro Magalhães, and W. Bas de Haas. A functional approach to automatic melody harmonisation. In *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design - FARM '13*, page 47. ACM Press, 2013.

[17] Kenichi Kurihara and Taisuke Sato. An Application of the Variational Bayesian Approach to Probabilistic Context-Free Grammars. 2004.

[18] Martin Lange and Hans Leiß. To CNF or not to CNF - An Efficient Yet Presentable Version of the CYK Algorithm. *Informatica Didactica*, 8:1–21, 2009.

[19] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA, 1983.

[20] Mark Levine. *The jazz theory book*. Sher Music, 1995.

[21] Eita Nakamura, Masatoshi Hamanaka, Keiji Hirata, and Kazuyoshi Yoshii. Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 276–280, 2016.

[22] Markus Neuwirth and Martin Rohrmeier. Towars a syntax of the Classical cadence. In *What is a Cadence*, pages 287–338. 2015.

[23] Martin Rohrmeier. A generative grammar approach to diatonic harmonic structure. *Proceedings SMC'07, 4th Sound andMusic Computing Conference*, (July):11–13, 2007.

[24] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 3 2011.

[25] Martin Rohrmeier and Ian Cross. Tacit tonality : Implicit learning of context-free harmonic structure. In *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009) Jyväskylä, Finland*, number Escom, pages 443–452, 2009.

[26] Keith Salley and Daniel T. Shanahan. Phrase Rhythm in Standard Jazz Repertoire: A Taxonomy and Corpus Study. *Journal of Jazz Studies*, 11(1):1, 2016.

[27] Daniel Shanahan and Yuri Broze. Diachronic Changes in Jazz Harmony: A Cognitive Perspective. *Music Perception: An Interdisciplinary Journal*, 31(1):32–45, 2013.

[28] Daniel Shanahan, Yuri Broze, and Richard Rodgers. A Diachronic Analysis of Harmonic Schemata in Jazz. In *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, pages 909–917, 2012.

[29] Stuart M Shieber, Yves Schabes, and Fernando C.N. Pereira. Principles and Implementation of Deductive Parsing. *Journal of Logic Programming*, 1993.

[30] Mark J. Steedman. A Generative Grammar for Jazz Chord Sequences. *Music Perception: An Interdisciplinary Journal*, 2(1):52–77, 1984.

[31] Mark J Steedman. The blues and the abstract truth: Music and mental models. *Mental models in cognitive science: essays in honour of Phil Johnson-Laird*, pages 305–318, 1996.

[32] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

HARMONIC SYNTAX IN TIME RHYTHM IMPROVES GRAMMATICAL MODELS OF HARMONY

Daniel Harasim[1] **Timothy J. O'Donnell**[2] **Martin Rohrmeier**[1]
[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne, Switzerland
[2] Department of Linguistics, McGill University, Canada
daniel.harasim@epfl.ch

ABSTRACT

Music is hierarchically structured, both in how it is perceived by listeners and how it is composed. Such structure can be elegantly captured using probabilistic grammatical models similar to those used to study natural language. They address the complexity of the structure using abstract categories in a recursive formalism. Most existing grammatical models of musical structure focus on one single dimension of music–such as melody, harmony, or rhythm. While these grammar models often work well on short musical excerpts, accurate analysis of longer pieces requires taking into account the constraints from multiple domains of structure. The present paper proposes abstract product grammars–a formalism which integrates multiple dimensions of musical structure into a single grammatical model–along with efficient parsing and inference algorithms for this formalism. We use this model to study the combination of hierarchically-structured harmonic syntax and hierarchically-structured rhythmic information. The latter is modeled by a novel grammar of rhythm that is capable of expressing temporal regularities in musical phrases. It integrates grouping structure and meter. The combined model of harmony and rhythm outperforms both single-dimension models in computational experiments. All models are trained and evaluated on a treebank of hand-annotated Jazz standards.

1. INTRODUCTION

Music is hierarchically organized, which is probably most evident in the structure of harmonic sequences. Grammatical models of music describe both local and non-local relations between musical objects such as notes or chords by assuming a latent hierarchical structure. Originally inspired by Schenkerian analysis and generative linguistics [9], grammatical models have been used in music theory [14, 19, 24, 25], computational musicology [1, 5, 6, 13, 16, 27], music information retrieval [3, 4, 12, 18, 26], and increasingly also music psychology [7, 20]. Consider for example the Jazz chord sequence C^6 D^7 Dm^7 G^7 C^6 of the A-part of the Jazz standard *Take the A-Train*. A hierarchical analysis of this sequence is shown in Figure 1a. The progression D^7 Dm^7 G^7 forms a *dominant phrase* inside the *tonic phrase* C^6 D^7 Dm^7 G^7 C^6, exhibiting a non-local harmonic relationship between the chords D^7 and G^7. The nesting of the phrases moreover illustrates the idea of how pieces can be decomposed into hierarchically-structured constituents (subtrees) which stand in part-whole relationship with one another [6]. Figure 2 displays a typical case of a non-local harmonic relation in Jazz harmony.

To analyze hierarchical harmonic structures, music theorists make use of many additional structural features such as melody, rhythm, voice-leading, and form, for disambiguation. From this perspective, the latent harmonic structure of a piece cannot be fully inferred from sequences of chord symbols alone. Most existing grammatical models of harmony, however, do not take these other domains of musical structure in account. In this paper, we propose a novel formalism that combines models of different musical features. The mathematical idea is similar to Coupled-context-free Grammars [17]. We extend that approach by a probabilistic model and apply the general construction to improve models of harmonic syntax by incorporating harmonic rhythm.

1.1 Problem Setting

Existing grammatical models of harmony typically do not capture how harmonic structure is laid out in time [21], as shown in Figure 1a. This analysis captures information such as the dependencies between different kinds of musical phrase (tonic, dominant, subdominant), ordering, and hierarchical constituency, but contains no information on the duration of chords. This paper extends models of harmonic syntax to include rhythmic structure illustrated in Figure 1b. This figure shows how the musical phrases in Figure 1a are laid out in time by progressively assigning constituents to a metrical grid consisting of eight measures. The inclusion of the metrical domain reveals previously hidden structure. In the first step, the root of the harmonic tree is assigned to the entire eight bars. In the second step, the tonic phrase is split into equal halves which are assigned to bars 1-4 and bars 5-8 of the metrical grid. In the third step, the second half of the piece is split into equal halves, introducing a V in the first part of the split and limiting the tonic scale degree to the second part. The fourth step, in contrast, splits the first half (measures 1–4) into two and assigns the second half of this split to the second half of the progression (measures 5–8). Measures 3 and 4 are said to be a *harmonic upbeat* to measures 5 and 6. In the following, we present an integrated model of harmony and phrase rhythm [22] that accounts for the structural differences of the steps three and four. Note that we therefore assume the existence of hypermeter, the extension of metrical structures within a single measure to relations between measures [11].

We propose an approach that models the upbeat and the downbeat of harmonic constituents separately. Figure 1c shows a hierarchical analysis integrating harmonic syntax and harmonic rhythm. In this notation, the durations of upbeats are separated from the durations of downbeats by the symbol $\oplus$. The symbol $\ominus$ is used to indicate the "time stealing" from generation step 3 in Figure 1b.

2. GRAMMATICAL MODELS

2.1 Abstract Context-Free Grammars

The following two definitions are adopted from [6], where further explanation and examples can be found.

A *(non-probabilistic) Abstract Context-free Grammar* $G = (T, C, C_0, \Gamma)$ consists of a set T of *terminal symbols*, a set C of *constituent categories*, a set of *start categories* $C_0 \subseteq C$, and a set of partial functions

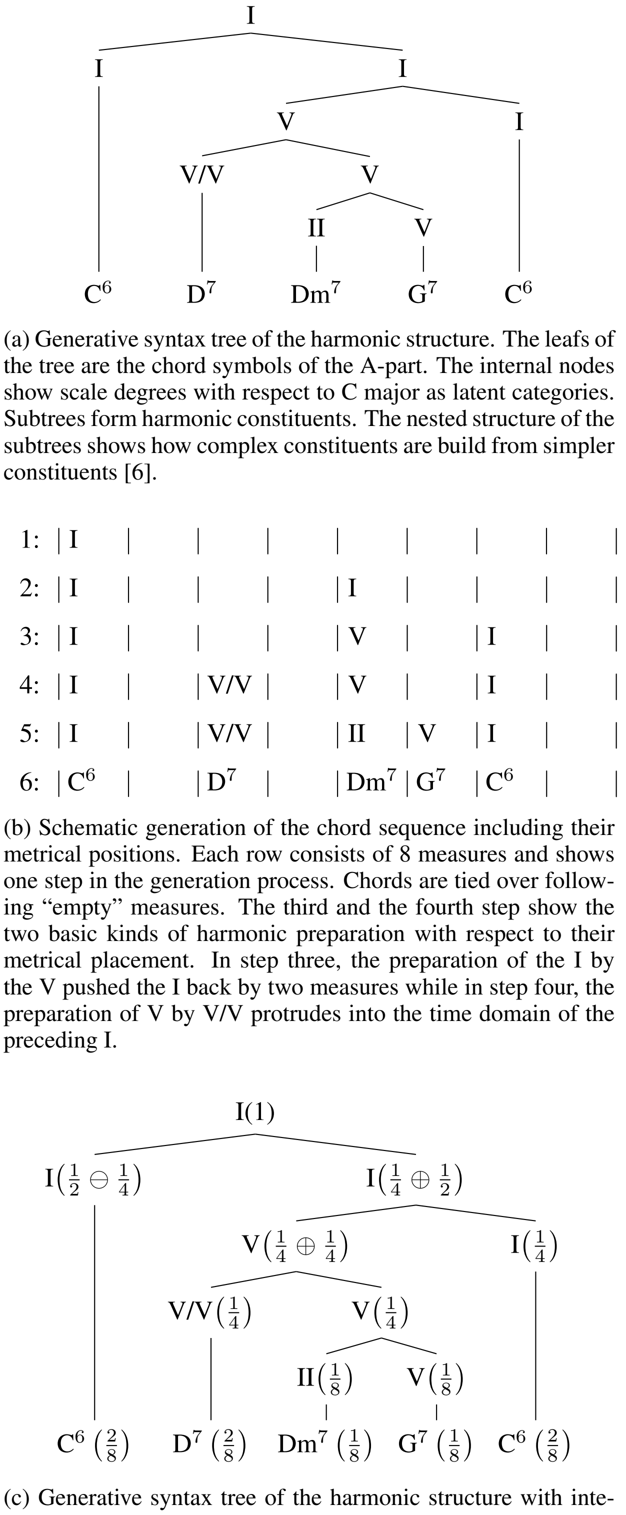
$$\Gamma := \{\, r \mid r : C \nrightarrow (T \cup C)^* \,\}, \qquad (1)$$

called *rewrite rules* or *rewrite functions*. The arrow $\nrightarrow$ is used throughout the paper to denote partial functions and $\mathrm{dom}(r)$ denotes the set of arguments for which a partial function r is defined. A sequence $\beta \in (T \cup C)^*$ can be generated from a sequence $\alpha \in (T \cup C)^*$ by one *rule application* of a rewrite function $r \in \Gamma$, denoted by $\alpha \longrightarrow_r \beta$, if there exist $\alpha_1, \alpha_2 \in (T \cup C)^*$ and $A \in C$ such that $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 r(A) \alpha_2$. A sequence of rewrite rules $r_1 \dots r_n$ is called a *derivation* of a sequence of terminals $\alpha \in T^*$ if there exists a start category $\alpha_1 \in C_0$, and $\alpha_2, \dots, \alpha_n \in (C \cup T)^*$ such that

$$\alpha_1 \longrightarrow_{r_1} \alpha_2 \longrightarrow_{r_2} \cdots \longrightarrow_{r_n} \alpha, \qquad (2)$$

where r_i is always applied to the leftmost category of α_i for $i \in \{1, \dots, n-1\}$. The set of derivations of α is denoted by $D(\alpha)$. The language of the grammar G is the set of terminal sequences that have a derivation in G.

A *Probabilistic Abstract Context-free Grammar* is an Abstract Context-free Grammar where each category $A \in C$ is associated with a random variable X_A over rewrite functions r such that the probability $\mathbb{P}(X_A = r)$ is positive if and only if $r(A)$ is defined, that is $A \in \mathrm{dom}(r)$. In the following, we also use the notation $p(r \mid A) = \mathbb{P}(X_A = r)$ and $p(A \longrightarrow_r \alpha) = \mathbb{P}(X_A = r)\ \mathbb{1}(r(A) = \alpha)$. The probability $p(d)$ of a derivation $d = r_1 \dots r_n$ of a sequence of terminal symbols $\alpha \in T^*$ is defined as the product $\prod_{i=1}^{n} \mathbb{P}(r_i \mid A_i)$ where in each step r_i is applied to a category $A_i \in C$. The probability of α is then defined as $p(\alpha) = \sum_{d \in D(\alpha)} p(d)$.



(a) Generative syntax tree of the harmonic structure. The leafs of the tree are the chord symbols of the A-part. The internal nodes show scale degrees with respect to C major as latent categories. Subtrees form harmonic constituents. The nested structure of the subtrees shows how complex constituents are build from simpler constituents [6].

1:	I							
2:	I				I			
3:	I				V		I	
4:	I		V/V		V		I	
5:	I		V/V		II	V	I	
6:	C^6		D^7		Dm^7	G^7	C^6	

(b) Schematic generation of the chord sequence including their metrical positions. Each row consists of 8 measures and shows one step in the generation process. Chords are tied over following "empty" measures. The third and the fourth step show the two basic kinds of harmonic preparation with respect to their metrical placement. In step three, the preparation of the I by the V pushed the I back by two measures while in step four, the preparation of V by V/V protrudes into the time domain of the preceding I.

(c) Generative syntax tree of the harmonic structure with integrated rhythmic information. The numbers in parentheses denote the duration of the constituents relative to the whole progression. The branch $\mathrm{I}(1) \longrightarrow \mathrm{I}(\frac{1}{2} \ominus \frac{1}{4})\ \ \mathrm{I}(\frac{1}{4} \oplus \frac{1}{2})$ is an instance of a split that anticipates the upbeat preparation of G^7 by D^7. Because of a 2 measures long upbeat, the left child is 2 measures shorter and the right child is 2 measures longer than in a preparation without an upbeat.

Figure 1: Hierarchical analysis of the A-part of the Jazz standard *Take the A-Train* in C major, considering the structural domains of harmony and rhythm.

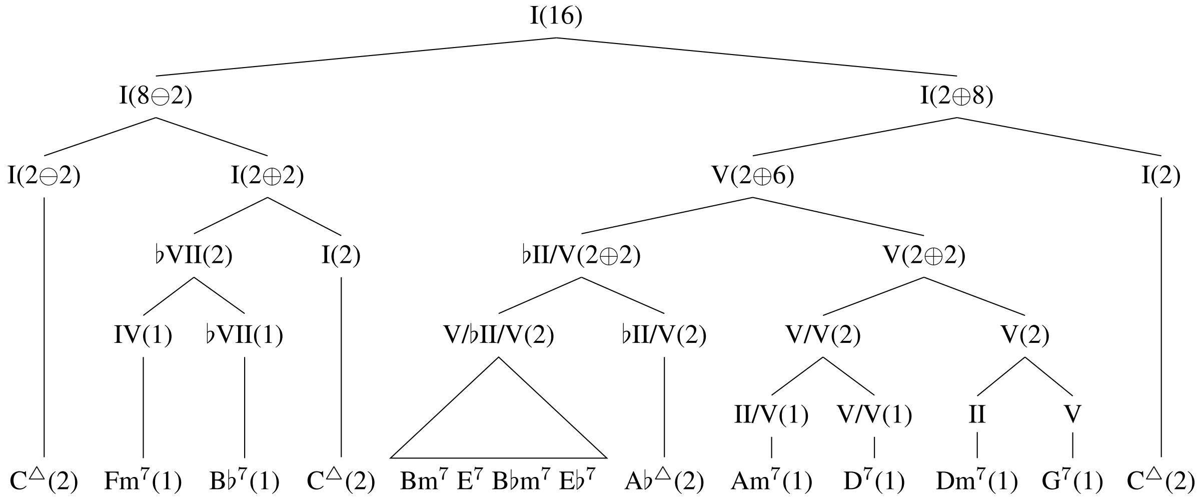


Figure 2: Hierarchical analysis of the Jazz standard *Half Nelson*, integrating harmonic and rhythmic structure. In this tree, a duration of 1 corresponds to one measure for the sake of readability (the whole tune spans 16 measures). The non-local dependency between the chords A$\flat^{\triangle}$ and G^7 constitutes a characteristic harmonic relation of the tune.

2.2 Product Grammars

This paper proposes to improve generative grammar models of harmony by forming a product of a harmony grammar and a rhythm grammar.

Let $G = (T, C, C_0, \Gamma)$ and $G' = (T', C', C_0', \Gamma')$ be two PACFGs and let $\mathrm{ar}(r)$ denote the *arity* of a rule r, which is defined as the length of its right-hand side. The *product grammar*

$$G \bowtie G' = (T \times T', C \times C', C_0 \times C_0', \Gamma \bowtie \Gamma') \quad (3)$$

is constructed from the Cartesian products of the sets of terminals, categories, and start categories. The rewrite functions of $G \bowtie G'$ are all pairs of functions of equal arity,

$$\Gamma \bowtie \Gamma' = \{ (r, r') \in \Gamma \times \Gamma' \mid \mathrm{ar}(r) = \mathrm{ar}(r') \}. \quad (4)$$

For a product category $(A, A') \in C \times C'$ and rewrite functions $r \in \Gamma$ and $r' \in \Gamma'$ of equal arity, the application of (r, r') to (A, A') is defined component-wise,

$$(r, r')(A, A') = (r(A), r'(A')). \quad (5)$$

By abuse of notation, the right-hand side of this equation does not stand for a pair of sequences, but a sequence of pairs. The probability of a product rule application is defined as the product of the probabilities of the rule application components,

$$p((r, r') \mid (A, A')) = p(r \mid A)\, p(r' \mid A'). \quad (6)$$

That is, the choice of rule r is set to be independent of A' and r', and the choice of r' is independent of A and r in the generative process.

A helpful intuition of product grammars is that they compute the intersection of two sets of derivation trees for a sequence. The derivation trees of the grammar $G \bowtie G'$ are exactly those which are derivations in both G and G' if the labels of the trees (terminals and categories) are ignored. The probability of a derivation in $G \bowtie G'$ is then also equal to the product of its corresponding derivations in G and G'.

2.3 Rhythm Grammar

2.3.1 *Full Rhythm Grammar*

A *rhythmic category* $a \oplus b$ consists of two rational numbers $a \in \mathbb{Q}$ and $b \in \mathbb{Q}$ such that $0 \leq a$, $0 < b$, and $a + b \leq 1$. The first number a is called the *upbeat* and the second number b is called the *downbeat* of the category. The intuition behind the symbol $\oplus$ is that the total length of a rhythmic category equals the sum of its two components, $\lambda(a \oplus b) := a + b$, where λ is the function that denotes the length of the rhythmic constituent as a proportion of the overall piece, which is fixed to be the unit $1 \in \mathbb{Q}$. The condition $0 \leq a$ forbids negative upbeat parts, $0 < b$ ensures positive category lengths, and $a + b \leq 1$ ensures that no category is longer than the whole piece. For convenience, we use two additional short-hand notations: a category with no upbeat is denoted by the length of its downbeat, $b = 0 \oplus b$. The category of a rhythmic constituent that loses a portion c of its downbeat (formerly with length b) to the upbeat of the following rhythmic constituent is denoted by $b \ominus c := 0 \oplus (b - c)$. In this case $\lambda(b \ominus c) = b - c$, too.

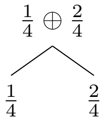
The start category of the rhythmic grammar is 1, the length of the piece, and any category with zero upbeat is allowed to be a terminal (leaf node). The essential grammar rules are given by two families of rewrite functions, one family of partial functions for splitting the upbeat components of categories $\mathtt{usplit}_v : C \nrightarrow C^*$ and one family of total functions for splitting the downbeats $\mathtt{dsplit}_v^u : C \to C^*$,

$$\mathtt{usplit}_u(a \oplus b) := ((1-u)a \oplus ua) \qquad (0 \oplus b) \quad (7)$$

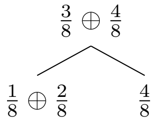
$$\mathtt{dsplit}_w^v(a \oplus b) := (a \oplus (1 - v - vw)b) \quad (vwb \oplus wb),$$

where $u, v, w \in \mathbb{Q}$ such that $\frac{1}{2} < u \leq 1$ and $a > 0$ in the first equation, and $0 \leq v < 1$ and $0 < w < 1$ in the second equation. The parameter u represents the downbeat proportion of the upbeat, v is the upbeat proportion of the second category of a downbeat split, and w is the downbeat proportion of the second category of a downbeat split.

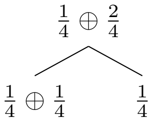
In other words: The upbeat split rule $\texttt{usplit}_u$ separates the upbeat from the downbeat and optionally splits the upbeat again into a new upbeat and downbeat. For example for $u = 1$ and $u = \frac{2}{3}$:



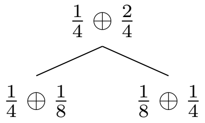
and


$\frac{3}{8} \oplus \frac{4}{8}$
$\frac{1}{8} \oplus \frac{2}{8}$
$\frac{4}{8}$

In contrast, the downbeat split $\texttt{dsplit}^v_w$ ignores the upbeat and splits the downbeat. It optionally introduces a new upbeat preparation. For example for $v, w = 0, \frac{1}{2}$ and $v, w = \frac{1}{2}, \frac{1}{2}$:


$\frac{1}{4} \oplus \frac{2}{4}$
$\frac{1}{4} \oplus \frac{1}{4}$
$\frac{1}{4}$

and


$\frac{1}{4} \oplus \frac{2}{4}$
$\frac{1}{4} \oplus \frac{1}{8}$
$\frac{1}{8} \oplus \frac{1}{4}$

One rule $\texttt{unary}(a \oplus b) := a \oplus b$ is added to the grammar to ensure compatability with grammars that use rewrite rules of arity one.

The probability of a rhythmic rewrite functions does not depend on the particular rhythmic category that it rewrites, but only on whether or not the category has an upbeat of length zero. This enables a maximal sharing of probability mass by preserving consistency with the constraints of the rewrite rules. More precisely,

$$\begin{aligned} 1 = \; & p(\texttt{unary} \mid a \oplus b) \\ & + \sum_{\frac{1}{2} < u \leq 1} p(\texttt{usplit}_u \mid a \oplus b) \\ & + \sum_{0 \leq v < 1} \sum_{0 < w < 1} p(\texttt{dsplit}^v_w \mid a \oplus b) \end{aligned} \tag{8}$$

for $a > 0$ and

$$\begin{aligned} 1 = \; & p(\texttt{unary} \mid 0 \oplus b) \\ & + \sum_{0 \leq v < 1} \sum_{0 < w < 1} p(\texttt{dsplit}^v_w \mid 0 \oplus b). \end{aligned} \tag{9}$$

For practical applications, the parameters u, v, and w are limited to a finite set of rational numbers to put a proper normalized prior on the rule distributions.

2.3.2 Simplified Rhythm Grammar

For comparison, we additionally consider a simplified version of the rhythm grammar presented above which does not explicitly model upbeats. The rhythmic categories and the terminals of this grammar are rational numbers $0 < a \leq 1$ representing constituent durations relative to the full piece. Apart from the technical unary rule, the rules of the grammar form a family of total rewrite functions

$$\texttt{split}_s(a) := (sa) \qquad (a - sa). \tag{10}$$

The parameter $0 < s < 1$ is called the *temporal split ratio* of the rule. The probabilities of the rewrite rules are set to be independent from the category they rewrite. Therefore,

$$1 = p(\texttt{unary}) + \sum_{a \in \mathbb{Q}} p(\texttt{split}_a). \tag{11}$$

2.4 Harmony Grammar

The harmony grammar used in this paper is a standard probabilistic context-free grammar (Σ, N, S, R) in Chomsky normal form. It consists of a set Σ of chord symbols as terminal symbols, a set of copies of chord symbols N as non-terminal symbols, a distinguished start symbol $S \in N$, and a set of standard rewrite rules

$$R \subseteq \{\, A \longrightarrow B_1\ B_2 \mid B_k \in N, A = B_1 \text{ or } A = B_2 \,\}\,.$$

In particular, rules of the form $A \longrightarrow A\ A$ are included by this definition. Each non-terminal symbol A is also associated with a random variable X_A over rewrite rules that have A as their left-hand side. The symbols, rules, and parameters of the grammar are read from dataset of tree annotations described in the next section.

Note that since every rewrite rule of a standard context-free grammar can be interpreted as a partial function with a singleton domain,

$$\text{dom}(A \longrightarrow \alpha) = \{\, A \,\} \quad \text{for all } \alpha \in (\Sigma \cup N)^*, \tag{12}$$

every standard context-free grammar is also an Abstract Context-free Grammar and can be used in the product grammar construction.

3. DATASET

This study uses a dataset of 75 hand-annotated tree analyses of Jazz chord sequences from the iRealPro dataset [23]. The tree annotations were performed by the authors and a student assistant. Each chord sequence is annotated with a single binary tree that spans the whole piece. In contrast to the introductory examples of this paper, the internal nodes of each tree in the data are not labeled by scale degrees but chord symbols. depth one subtrees corresponds to a rule of the grammar described in the previous section. Figure 3 shows the absolute frequencies of the 20 most frequent harmonic rewrite rules from the dataset, after each sequence was transposed to the root of C. Rules of the form $A \longrightarrow A \quad A$, called *prolongation rules*, and rules of the form $A \longrightarrow B \quad A$ for $A \neq B$, called *preparation rules*, are the most used rule schemes.

The dataset additionally includes the length of each chord in quarter notes. The chord durations of each piece are divided by the total duration of the piece. From the chord durations and the harmonic tree annotations, the duration of each constituent (subtree) can be calculated automatically as shown in Figure 4. The temporal split ratios of the rule applications–as introduced in Equation 10–are then in turn calculated from the durations of the constituents. Consider for example the rule application $\text{G}^7(\frac{5}{32}) \longrightarrow \text{F}^{\triangle}(\frac{2}{32}) \;\; \text{G}^7(\frac{3}{32})$ from Figure 4. Its temporal split ratio is $\frac{2}{5}$.

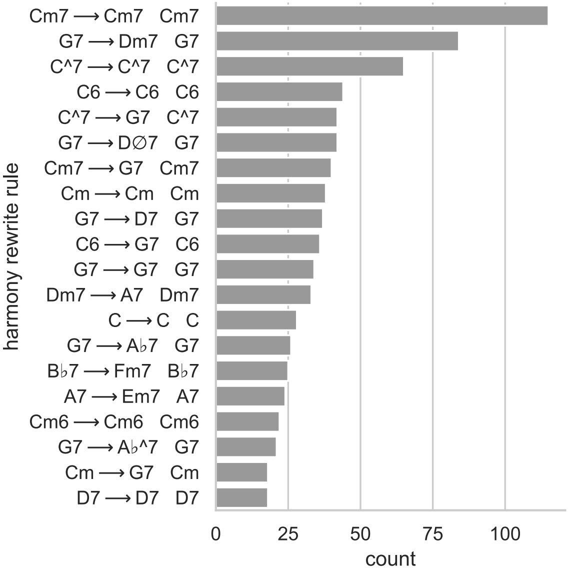


Figure 3: Absolute frequencies of the 20 most frequent harmonic rewrite rules of the tree annotations. All sequences are transposed to the common root C. Major-seventh chords are denotes as `C^7` and `A♭^7`.

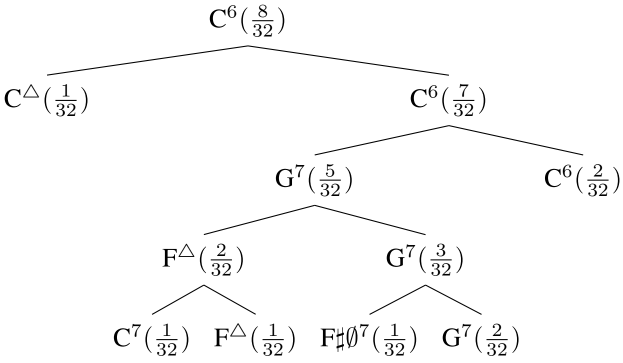


Figure 4: Tree annotation of the last chords of *St. Thomas*. Chord durations are shown relative to the total duration of the tune, $\frac{2}{32}$ corresponds to one measure. The durations of the inner nodes are calculated automatically.

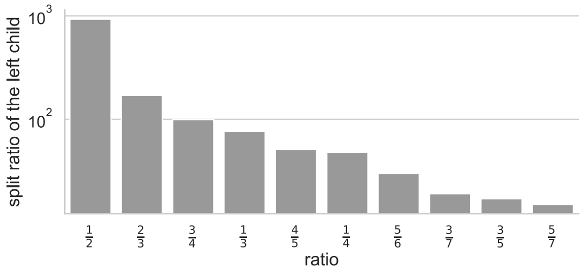


Figure 5: Absolute frequencies of the 10 most frequent split ratios of annotated tree constituents. The split ratio of a binary rewrite rule is defined as the time proportion of the left child. The y-axis is plotted using a logarithmic scale.

The 10 most frequent temporal split ratios are shown in Figure 5. The split ratio $\frac{1}{2}$ is by far the most frequent one. Most of the remaining ratios can be expressed either as $\frac{n-1}{n}$ or as $\frac{1}{n}$ for some $n \in \mathbb{N}$. The former arise for example from chains of descending fifths or applied dominants that accumulate time step by step in the temporal order of the piece. The latter arise from upbeat preparations that can be understood using the rhythmic categories described in Section 2.3.1. Two rhythmic rewrite rules that explain a split ratio of $\frac{1}{n}$ are $\left(\frac{n}{m}\right) \longrightarrow \left(\frac{\frac{n}{2}}{m} \ominus \frac{\frac{n}{2}-1}{m}\right) \quad \left(\frac{\frac{n}{2}-1}{m} \oplus \frac{\frac{n}{2}}{m}\right)$ and $\left(\frac{1}{m} \oplus \frac{n-1}{m}\right) \longrightarrow \left(\frac{1}{m}\right) \quad \left(\frac{n-1}{m}\right)$, where $m \in \mathbb{N}$. The former results from a downbeat split with $w = \frac{1}{2}$ and the latter results from an upbeat split with $u = 1$.

4. PARSING WITH PRODUCT GRAMMARS

A naive approach to parsing against a product grammar would enumerate all product categories and memoize the inverted rewrite rules on these categories. In this section, we show how the inefficient blow-up of the number of categories can be avoided using the independence assumption of Equation 6.

Consider an Abstract Context-Free Grammar in Chomsky normal form. The standard CYK algorithm–here used to calculate the probability of a sequence of terminals $w \in T^*$ of length n, indexed from 0 to $n-1$–can be formulated recursively by the equations

$$p(A, i, i) = \sum_{r \in \Gamma} p(A \longrightarrow_r w_i) \tag{13}$$

and

$$\begin{aligned} &p(A, i, j) \qquad (14) \\ &= \sum_{k=i}^{j-1} \sum_{r \in \Gamma} \quad p(A \longrightarrow_r B_1\ B_2) p(B_1, i, k)\, p(B_2, k+1, j) \end{aligned}$$

where $A, B_1, B_2 \in C$ and $i, j \in \mathbb{N}$ such that $0 \leq i < j \leq n-1$. The probability of the sequence is then given by $p(w) = \sum_{A \in C_0} p(A, 0, n-1)$.

Given a product grammar $G \bowtie G'$, a sequence of product terminals can be parsed utilizing Equation 6,

$$p((A, A'), i, i) = \sum_{(r, r') \in \Gamma \bowtie \Gamma'} p(A \longrightarrow_r w_i)\, p(A' \longrightarrow_{r'} w'_i) \tag{15}$$

and

$$\begin{aligned} p((A, A'), i, j) = \sum_{k=i}^{j-1} \sum_{(r, r') \in \Gamma \bowtie \Gamma'} p(A \longrightarrow_r B_1\ B_2) \qquad (16) \\ p(A' \longrightarrow_{r'} B'_1\ B'_2) p((B_1, B'_1), i, k)\, p((B_2, B'_2), k+1, j) \end{aligned}$$

It is therefore sufficient to parse the component grammars individually at each step. In other words, the combined grammar is computed on-the-fly to achieve efficiency.

5. EXPERIMENTS

We compare four product grammars that integrate harmonic and rhythmic structure. Additionally, we report the performances of their single-domain components and of a random baseline. As first component, we consider the harmony grammar presented in Section 2.4, trained either on the annotations in the original keys of the tunes or on the annotations after each tune was transposed to C major. As second component, we consider the full rhythm grammar presented in Section 2.3.1 that distinguishes upbeats and downbeats of constituents, and its simplification that uses the total length of the constituents, presented in Section 2.3.2. All models are trained and evaluated on the dataset described in Section 3. Apart from the full rhythm grammar, all models are trained by counting the harmonic rewrite rules or the temporal split ratios present in the dataset. The full rhythm grammar is trained using variational Bayesian inference [8]. Every model predicts the latent tree structure of a given sequence using the maximum a posteriori tree. One-fold cross validation was applied to avoid overfitting to the data: 75 times the model was trained on 74 sequences and evaluated on the remaining sequence.

5.1 Evaluation Metric and Baseline

The similarity of two trees is calculated as the unlabeled tree accuracy, defined as follows. Let α be a sequence of n terminals, left-to-right indexed from 0 to $n-1$, let t be a tree with α as leafs, and let s be a subtree of t. The *span* of s is defined as the pair of the index of its left-most child and the index of its right-most child. The set of spans of t consists of the spans of all subtrees of t that are not leafs. The unlabeled tree accuracy of a tree prediction t to the respective Goldstandard tree t^* is then defined as the cardinality of the correctly predicted spans, divided by the total amount of spans of t^*.

Given a chord sequence of length n, the random baseline uniformly samples one tree from the set of all binary trees with n leafs.

5.2 Results and Discussion

The results of the computational experiments are shown in Figure 6. All combined models of harmony and rhythm perform significantly better than the single-domain harmony grammars and all models perform significantly better than the random baseline ($p < 0.01$ using 2-sample bootstrap tests). There is no statistical difference observable between the not transposed and the transposed harmony models. Surprisingly, the single-domain rhythm grammars perform much better than the single-domain harmony grammars. This is, however, only possible because we consider the unlabeled tree accuracy. Other measures such as perplexity would reveal the obvious incapability of the rhythm grammars to predict chord sequences.

Both rhythm grammars improve the harmony models similarly. As discussed in Section 3, the simplified version of the proposed rhythm grammar is also able to capture some complex rhythmical structures. The music-theoretically more sophisticated formalism, however, facilitates the interpretation and explanation of the observed split ratios.

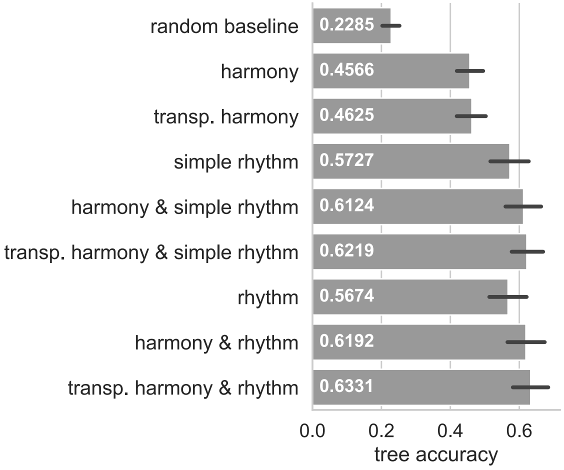


Figure 6: One-fold cross-validated tree accuracies of the tested models and the random baseline. The error bars show 95% bootstrap confidence intervals. The combined models of harmony and rhythm perform significantly better than the plain harmony grammars.

6. CONCLUSION

The usage of rhythmical information is shown to significantly improve the performance of harmonic syntax models. The empirical comparison between a music-theoretical motivated model and its simplified version shows that both models improve the harmony grammar equally well. The simplified model can therefore be used as an algorithmic proxy of the more expressive model. This might, however, only be true for rhythmically regular structures such as the harmonic rhythm of chord sequences from Jazz standards. It is, moreover, surprising how much information is already contained in the rhythm of the sequences, which underpins the importance of the rhythmic dimension of music [10]. In these sequences, both the harmonic syntax and the phrase rhythm work together to strengthen the intentionality of the music.

The here proposed model of interaction between harmony and rhythm is also capable to describe the interaction of pitch and rhythm in melodies. A rewrite function for syncopation could be added for future applications, since syncopation is an essential part of melodic rhythm.

The general product grammar construction presented in this paper integrates multiple domains of structure using strong independence assumptions. Future research can extent the formalism, explicitly modeling inter-domain dependencies. We hope that the presented approach will prove to be useful for applications such as rhythm quantization [2], the definition of similarity metrics [5], and computational composition assistance [15].

7. ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 – PMSB and from the Natural Sciences and Engineering Research Council of Canada (NSERC). We thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology. The authors additionally thank the anonymous referees for their valuable comments.

8. REFERENCES

[1] Samer A. Abdallah and Nicolas E. Gold. Comparing models of symbolic music using probabilistic grammars and probabilistic programming. *Proceedings of ICMC/SMC*, pages 1524–1531, 2014.

[2] Francesco Foscarin, Florent Jacquemard, Philippe Rigaux, and Sakai Masahiko. A parse-based framework for coupled rhythm quantization and score structuring. In *Mathematics and Computation in Music*, pages 248–260, Cham, 2019. Springer International Publishing.

[3] Mark Granroth-Wilding and Mark Steedman. A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43(4):355–374, October 2014.

[4] W Bas De Haas, José Pedro Magalhães, and Frans Wiering. Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 295–300, 2012.

[5] W Bas De Haas, Martin Rohrmeier, and Frans Wiering. Modeling Harmonic Similarity using a Generative Grammar of Tonal Harmony. *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, 2009.

[6] Daniel Harasim, Martin Rohrmeier, and Timothy J O'Donnell. A Generalized Parsing Framework for Generative Models of Harmonic Syntax. *19th International Society for Music Information Retrieval Conference*, 2018.

[7] Stefan Koelsch, Martin Rohrmeier, Renzo Torrecuso, and Sebastian Jentschke. Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, 110(38):15443–15448, 2013.

[8] Kenichi Kurihara and Taisuke Sato. An Application of the Variational Bayesian Approach to Probabilistic Context-Free Grammars. In *International Joint Conference on Natural Language Processing {(IJCNLP-04)} Workshop Beyond Shallow Analyses*, 2004.

[9] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA, 1983.

[10] Florence Levé, Richard Groult, Guillaume Arnaud, Cyril Séguin, Rémi Gaymay, and Mathieu Giraud. Rhythm extraction from polyphonic symbolic music. In *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pages 375–380, 2011.

[11] Ryan McClelland. Extended upbeats in the classical minuet: Interactions with hypermeter and phrase structure. *Music Theory Spectrum*, 28(1):23–55, Spring 2006.

[12] Andrew McLeod and Mark Steedman. Meter Detection in Symbolic Music Using a Lexicalized PCFG. *Proceedings of the 14th Sound and Music Computing Conference*, 2017.

[13] Eita Nakamura, Masatoshi Hamanaka, Keiji Hirata, and Kazuyoshi Yoshii. Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 276–280, 2016.

[14] Markus Neuwirth and Martin Rohrmeier. Towards a syntax of the Classical cadence. In *What is a Cadence*, pages 287–338. Leuven University Press, 2015.

[15] Alexandre Papadopoulos, Pierre Roy, and François Pachet. Assisted lead sheet composition using flowcomposer. In Michel Rueher, editor, *Principles and Practice of Constraint Programming*, pages 769–785, Cham, 2016. Springer International Publishing.

[16] Marcus Pearce and Martin Rohrmeier. Musical syntax ii: empirical perspectives. In *Springer handbook of systematic musicology*, pages 487–505. Springer, 2018.

[17] Gisela Pitsch. LR(k)-Parsing of Coupled-Context-Free Grammars. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, 1994.

[18] Donya Quick. Learning production probabilities for musical grammars. *Journal of New Music Research*, 45(4):295–313, 2016.

[19] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.

[20] Martin Rohrmeier and Ian Cross. Tacit tonality: Implicit learning of context-free harmonic structure. In *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009) Jyväskylä, Finland*, pages 443–452, 2009.

[21] Martin Rohrmeier and Marcus Pearce. Musical syntax i: Theoretical perspectives. In *Springer handbook of systematic musicology*, pages 473–486. Springer, 2018.

[22] Keith Salley and Daniel T. Shanahan. Phrase Rhythm in Standard Jazz Repertoire: A Taxonomy and Corpus Study. *Journal of Jazz Studies*, 11(1):1, 2016.

[23] Daniel Shanahan, Yuri Broze, and Richard Rodgers. A Diachronic Analysis of Harmonic Schemata in Jazz. In *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, pages 909–917, 2012.

[24] Mark J. Steedman. A Generative Grammar for Jazz Chord Sequences. *Music Perception: An Interdisciplinary Journal*, 2(1):52–77, 1984.

[25] Mark J Steedman. The blues and the abstract truth: Music and mental models. *Mental models in cognitive science: essays in honour of Phil Johnson-Laird*, pages 305–318, 1996.

[26] Hiroaki Tsushima, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Function- and Rhythm-Aware Melody Harmonization Based on Tree-Structured Parsing and Split-Merge Sampling of Chord Sequences. *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, pages 205–208, 2017.

[27] Hiroaki Tsushima, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Generative statistical models with self-emergent grammar of chord sequences. *Journal of New Music Research*, 47(3):226–248, May 2018.

THE JAZZ HARMONY TREEBANK

Daniel Harasim[1] **Christoph Finkensiep**[1] **Petter Ericson**[1]
Timothy J. O'Donnell[2] **Martin Rohrmeier**[1]
[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne, Switzerland
[2] Department of Linguistics, McGill University, Canada
daniel.harasim@epfl.ch

ABSTRACT

Grammatical models which represent the hierarchical structure of chord sequences have proven very useful in recent analyses of Jazz harmony. A critical resource for building and evaluating such models is a ground-truth database of syntax trees that encode hierarchical analyses of chord sequences. In this paper, we introduce the Jazz Harmony Treebank (JHT), a dataset of hierarchical analyses of complete Jazz standards. The analyses were created and checked by experts, based on lead sheets from the open iRealPro collection. The JHT is publicly available in JavaScript Object Notation (JSON), a human-understandable and machine-readable format for structured data. We additionally discuss statistical properties of the corpus and present a simple open-source web application for the graphical creation and editing of trees which was developed during the creation of the dataset.

1. INTRODUCTION

Jazz music exhibits hierarchical relations between chords. This is particularly apparent in the fact that virtually any chord of a Jazz standard can be prepared by an applied dominant or subdominant. In fact, many chord sequences can be explained as the recursive application of such preparations [40]. Chords that are far apart in time can therefore be directly related, establishing long-range dependencies that can span whole formal sections of pieces. Such hierarchical structures also correlate with empirical findings from music perception research [25]. This is by no means to say that hierarchies are the only relevant relations between chords. Hierarchical chord relations are, however, underrepresented in computational models of harmony to date; the here presented dataset is intended to ease the development of hierarchical models.

Inspired by Schenkerian theory [3, 44] and generative syntax formalisms for natural language, generative theories of harmonic syntax model the hierarchical relations in chord sequences based on formal grammatical devices such as context-free grammars. Recent research uses formal grammars to represent hierarchical relations in melodies [1,9,12,15,24,33], chord sequences [14,19,42], and rhythms [18,29]. The fields of application include music theory [36, 39], music psychology [25, 41], automatic harmonic analysis [7, 16], and automatic music transcription [10,30,34].

The aim of this article is to present the Jazz Harmony Treebank (JHT), a dataset of hierarchical harmonic analyses of Jazz standards by music experts in a human-understandable and machine-readable format. We report on the creation of the treebank, describe the musical interpretation of the syntax trees, and explain the decisions that were made to meet the challenges of the annotation procedure. The dataset is available on GitHub. [1]

Treebanks are of particular importance for the study of hierarchical models and their applications. In linguistics, they have been and remain instrumental for many natural language processing tasks. The well-known Penn Treebank [28], first published in the early nineties, is an instructive example since it has been used as an object of study in and of itself [11], as a basis for publishing additional treebanks with different paradigms [21] and for different languages [27], and–most prominently–as a dataset for training and evaluating machine-learning methods [22,31,43].

The present article describes the creation process of the JHT. We take this as an opportunity to study the details of harmonic syntax using several concrete examples of Jazz standards. The major challenge of this application lies in the many individual decisions analysts have to take to address the ambiguity of music. Importantly, our goal is not to create uniform syntax trees of Jazz chord sequences, but to describe individual and subjective listening experiences in an unambiguous formal representation. Harmonic relations in sufficiently long chord sequences can be perceived in several ways, without one interpretation being clearly preferable. Therefore, the syntax trees of the JHT are best understood as proposals with a clear interpretation. The trees provide a basis for further analytical discussions, sophisticated computational models, and for education.

1.1 Related Symbolic Datasets

Many existing collections of symbolic data about chord sequences concentrate on providing chord labels for harmonic entities. Two prominent datasets of time-aligned

[1] https://github.com/DCMLab/JazzHarmonyTreebank

chord symbols were created by Harte et al. [20] and Burgoyne et al. [2] to study automatic chord transcription from audio. Neuwirth et al. [35] take a more music-theoretically motivated approach by proposing a chord-symbol representation for Western classical music and apply it to scale degree analyses of Beethoven's string quartets. Chen and Su [5] and Devaney et al. [8] similarly label excerpts of sonatas, madrigals, chorals, preludes, and songs from common-practice tonality. Micchi et al. [32] combine existing Roman numeral analyses into a meta-dataset.

The datasets just mentioned use chord labels to analyze music given as audio data or in a symbolic representation. Since we analyze the relations between the chords of such sequences, this study is located at a higher level of abstraction. Only a few datasets of hierarchical analyses of sequential musical data are available in divergent formats [38]. Hamanaka et al. [17] and Kirlin [23] created two datasets of tree analyses of melodies of Western Classical Music. Gotham and Ireland [13] study musical form by the creation of datasets in a hierarchical representation. Granroth-Wilding and Steedman [14] provide a dataset of 76 sub-sequences of Jazz standards with partial harmonic grouping labels. In contrast to previous research that analyzed snippets of musical pieces, the JHT consists of 150 full chord sequences of Jazz standards with complete harmonic syntax trees.

2. HARMONIC SYNTAX

A harmonic syntax tree, as shown in Figure 1a, denotes a mental representation of a musical piece as a whole. Unlike sequential models that describe how, for instance, a sequence of chord symbols is generated chord by chord from the start to the end, hierarchical models describe how the skeleton of a piece is generated and recursively elaborated [42]. In Jazz, the most prominent of those elaboration operations are the duplication of chords and the preparation of a chord by an applied dominant. Each application of an operation establishes a direct relation between two chords. A syntax tree consists exactly of the sum of all those relations. It is therefore not directly a model for first-time listening of a musical piece, but rather for the abstract representation of musicians or listeners who are (implicitly or explicitly) aware of a piece's harmonic relations. This usage of the word *syntax* is closely related to generative syntax formalisms of natural language that address the question of which relations between words a listener must notice to understand the meaning of a sentence [6].

The scope of this paper is limited to tonal Jazz, including Swing, Bossa Nova, Jazz Blues, Bebop, Cool Jazz, and Hard Bop, and excluding parts of traditional Blues, Modal Jazz, Free Jazz, and Modern Jazz. We furthermore excluded tunes such as *Groovin' High* whose harmonic structure requires even more expressive representations than trees.[2] The general idea of harmonic syntax is, however, also applicable to other musical styles such as Western classical music.

2.1 Prolongation and Preparation as Fundamental Principles

In the following, we present the syntactic formalism with a particular emphasis on its musical interpretation. The concept of functional harmony describes an expectation-realization structure between musical objects such as notes, chords, and keys. Consider for example the chords of the final cadence of the Jazz standard *Birk's Works*, Fm6 Abm7 Db7 G%7 C7 Fm6, where G%7 denotes a half-diminished seventh chord with root G. Figure 1b shows the expectation-realization structure of this chord sequence. The first Fm6 establishes the tonic and as such creates the expectation that the progression ends with Fm6. The chords Abm7 and Db7 function as the tritone-substituted subdominant and dominant of C7, respectively. They therefore create expectation that resolves in the (temporally distant) chord C7. The chord G%7 functions as a subdominant chord in F minor. It therefore creates expectation that resolves with the dominant chord C7 which itself resolves into the last tonic chord Fm6. We say that the tonic chords constitute a *prolongation*. The subdominant chords *prepare* the dominant chords and the dominant chords *prepare* the tonic chord. Abstractly, we say that a chord X *refers* to a chord Y if X either prolongs or prepares Y.[3]

Prolongation and preparation are the two fundamental principles of functional harmonic syntax [40]. They can be formalized as rules of a context-free grammar with chord symbols both as terminals and nonterminals. In the formalization, *strong prolongations* that prolong chords of the same root and chord form are distinguished from *weak prolongations* that prolong a chord with a functionally equivalent chord (e.g., prolongation of C with Am). Note that this concept of weak prolongation is more general than in the GTTM where prolonging chords are for instance required to have the same root [26]. Strong prolongation is represented by rules of the form $X \longrightarrow X\ X$ for chord symbols X (e.g., Fm6 $\longrightarrow$ Fm6 Fm6). For chord symbols X and Y, rules of the form $X \longrightarrow Y\ X$ and $X \longrightarrow X\ Y$ represent weak prolongations if X and Y are functionally equivalent (e.g., Fm6 $\longrightarrow$ Ab Fm6). If otherwise X and Y are not functionally equivalent, $X \longrightarrow Y\ X$ represents a preparation (e.g., Fm6 $\longrightarrow$ C7 Fm6).

The practise of having no separate alphabet of nonterminal symbols, and requiring each binary rule to have a left-hand side symbol also on the right-hand side, is related to dependency grammars [37] and categorical grammars [46] which are well-known in computational linguistics and natural language processing. The symbol that appears both on the left-hand side and the right-hand side is called the *head* of the rule. In our setting of prolongation and preparation, the prolonged (resp. prepared) chord is the head. Therefore, weak prolongation rules may be left- or right-headed, while preparation rules are always right-headed. In sum, our harmony grammar consists of the following rules which model strong prolongation, weak pro-

[2] *Groovin' High* exhibits crossing harmonic dependencies between a tonic prolongation from m1 to m5 and a dominant preparation from m4 to m7. A similar tune is *Out of Nowhere*.

[3] In contrast to models based on the *Generative Theory of Tonal Music* [26], we exclude the concept of departure as a primitive relation, because it is not consistent with our formalization of the expectation-realization structure.

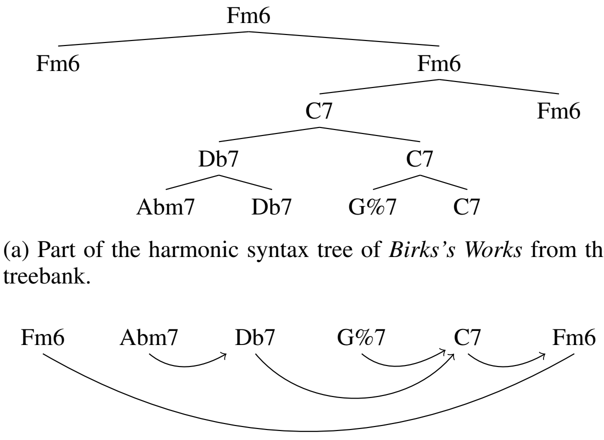
longation, and preparation, respectively,

$X \longrightarrow X\ X$	for any chord X	(strong prol.)
$X \longrightarrow Y\ X \mid X\ Y$	for any chord X and a functionally equivalent chord Y	(weak prol.)
$X \longrightarrow Y\ X$	for any chord X and a chord Y that prepares X	(preparation)

The tree in Figure 1a is a parse tree of the chord sequence Fm6 Abm7 Db7 G%7 C7 Fm6 under such a grammar of harmonic structure. Those parse trees represent exactly the same information as expectation-realization structures such as shown in Figure 1b: Undirected edges correspond to strong prolongations and directed edges correspond to either weak prolongations or preparations. This short example is unambiguous–it has only one plausible syntactic structure. In general, however, there are many syntax trees possible for a chord sequence. Grammar rules and syntax trees can then be weighted by probabilities that capture the plausibility of an analysis [1, 19, 24]. To identify the syntax tree that most accurately describes one's perception of the harmonic structure, other dimensions such as rhythm, form, and melody must also be taken into account. Even the artistic interpretation of a musical performance and the individual musical background of listeners have the potential to influence the perceived harmonic structure of a piece. A formal grammar that purely models chord symbols can therefore only answer the question "Is this a plausible syntax tree for a Jazz standard?", but not the question "Is this tree a good analysis of that particular tune in a particular context?". Until more complete models of musical structure are developed that integrate all relevant musical dimensions, the second question can only be answered by humans.

2.2 Complete Constituents and Open Constituents

Constituents formalize the notion of a musical unit such as a chord or a phrase. In the syntax tree shown in Figure 1a, the complete constituents are exactly the subsequences that are leafs of single subtrees, such as the subsequence Abm7 Db7 G%7 C7. Formally, we call a subsequence a *complete constituent* if it contains a chord, called the *head*, that is transitively referred to by all other chords of the sequence.[4] For instance, the chord C7 is the head of the phrase Abm7 Db7 G%7 C7 and Fm6 is the head of the whole sequence Fm6 Abm7 Db7 G%7 C7 Fm6. In cases in which a constituent is constituted by a strong prolongation (e.g., for the whole sequence of this example), we use the convention that the head is the right chord symbol. Since only the head of a complete constituent is allowed to refer to a chord outside the constituent, the concept of expectation-realization references is generalizable to complete constituents: we say that a complete constituent refers to a chord X if its head refers to X.

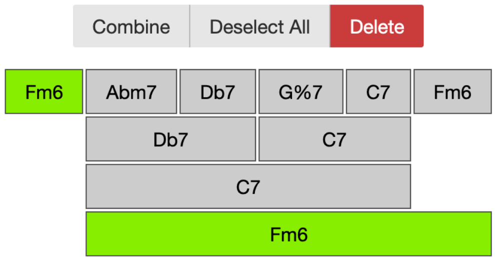


(a) Part of the harmonic syntax tree of *Birks's Works* from the treebank.

(b) Harmonic expectation-realization structure. This graph stands in 1-to-1 relation to the syntax tree shown in (a). Directed and undirected edges denote preparations and prolongations, respectively.

```
[.Fm6
    Fm6
    [.Fm6
        [.C7
            [.Db7
                Abm7
                Db7 ]
            [.C7
                G\%7
                C7 ] ]
        Fm6 ] ]
```

(c) String representation of the syntax tree in tikz-qtree format. This string is created using the tree annotation app shown in (d). The tree plot is shown in (a).



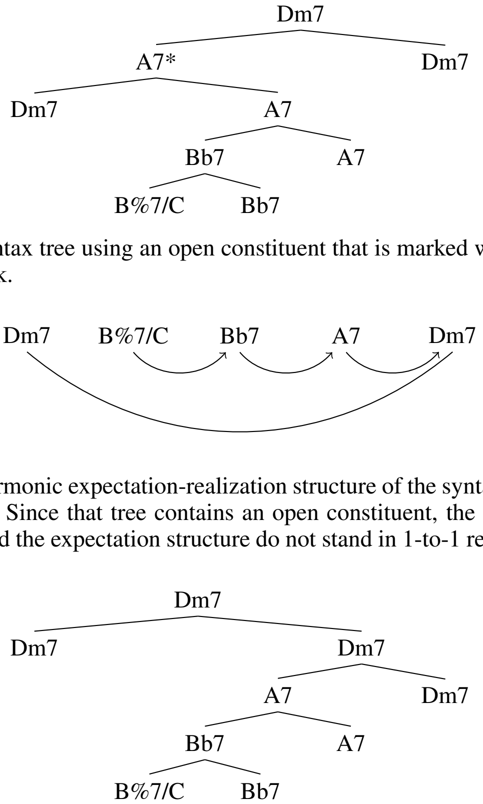
(d) Screenshot of tree annotation app. Each button represents a tree node. The user is selecting the green buttons to combine them to the full tree.

```
{"label": Fm6, "children": [
  {"label": "Fm6", "children": []},
  {"label": "Fm6", "children": [
    {"label": "C7", "children": [
      {"label": "Db7", "children": [
        {"label": "Abm7", "children": []},
        {"label": "Db7", "children": []}]},
      {"label: "C7", "children": [
        {"label: "G%7", "children": []},
        {"label: "C7", "children": []}]}]},
    {"label": "Fm6", "children": []}]}]}
```

(e) Tree string in JSON format, automatically converted from tikz-qtree format shown in (c).

Figure 1: Syntax tree of the final chords of the Jazz standard *Birk's works* in different representations.

[4] Note that the word head is used both for rules and constituents. This is not a problem since the head of a constituent is always the head of the top-most rule of its (sub-)tree analysis.



(a) Syntax tree using an open constituent that is marked with an asterisk.

(b) Harmonic expectation-realization structure of the syntax tree in (a). Since that tree contains an open constituent, the syntax tree and the expectation structure do not stand in 1-to-1 relation.

(c) Resolution of the open constituent in the syntax tree shown in (a). This tree stands in 1-to-1 relation to the expectation-realization structure in (b).

Figure 2: Hierarchical analysis of the initial chords of the Jazz standard *Why Don't You Do Right?* using open constituents (marked with asterisks).

In addition to complete constituents, one other constituent type is used in the JHT analyses. Consider for example the first four measures of the Jazz standard *Why Don't You Do Right?*,

| Dm7 B%7/C | Bb7 A7 | Dm7 B%7/C | Bb7 A7 |,

where B%7/C denotes a half-diminished seventh chord with root B and a C in the bass. The first two measures constitute a phrase following the *Lamento* schema (a stepwise descending movement of the bass from scale degree I to scale degree V [4]) that is repeated multiple times in the song. Since the transition from A7 to Dm7 does not sound like a resolution but more like a jump or an interruption (partly because of the repetition of the first two measures), we assume that A7 does not resolve into the following tonic Dm7, but into a tonic later in the song. Therefore, the phrase Dm7 B%7/C Bb7 A7 does constitute some kind of unit as shown in Figure 2a.

Since Dm7 and A7 both refer to a chord outside the phrase (see Figure 2b), the phrase does not have a head. It is therefore not a complete constituent. We call such constituents, in which multiple chords refer to a chord outside of the phrase, *open constituents*. The chords of an open constituent that refer to a chord outside of the constituent are called *chords with open references*. In the example of *Why Don't You Do Right?*, the chords Dm7 and A7 are the chords with open references of the open constituent Dm7 B%7/C Bb7 A7. Both chords Dm7 and A7 refer to the same tonic chord Dm7.

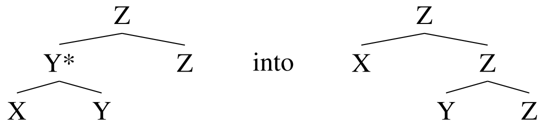
The JHT allows a single type of open constituent, called *restricted* open constituent, which consists of two adjacent constituents that refer to the same chord later in the piece. Since all constituents considered in the JHT are restricted in that way, we simple refer to them as open constituents. The restriction enables a further generalization of expectation-realization references to open constituents: We say that an open constituent refers to the chord to which all of its chords with open references refer. As shown in Figure 2a, the topmost node of an open constituent is labeled by the chord symbol of the right child of the node and additionally marked with an asterisk.

Other examples of open constituents are (i) I-VI-II-V-like phrases in *I Got Rhythm* and *I Can't Give You Anything But Love* and, in particular, (ii) tunes of form ABAC in which the B-part ends in a half cadence such as *All of Me*, *How High the Moon*, and *A Fine Romance*. *Summertime*, shown in Figure 3, is a prototypical example of a song with a ABAC form and a half cadence at the end of the B section. The interruption after the half cadence is supported by the movement from scale degree 3 to scale degree 2 in the melody and denoted using an open constituent.

2.3 Interpretation of Open Constituents as Prolongation-Preparation Structures

Syntax trees containing open constituents are interpretable as expectation-realization structures as shown in Figure 2. The interpretation procedure transforms a syntax tree that contains open constituents (e.g., Figure 2a) in to a tree that only represents prolongation and preparation operations (e.g., Figure 2c). This transformed tree then characterizes the expectation-realization structure (e.g., Figure 2b). Since open constituents are explicitly marked with asterisks, their interpretation is unambiguous.

To formalize the interpretation of open constituents, let Y^* be the chord symbol labeling an open constituent consisting of two constituents labeled with chord symbols X and Y. Let further be Z the chord symbol that is referenced by both X and Y. The reference is expressed by Z being the right sibling of the open constituent. The conversion then transforms



In the more general case of nested open constituents, the conversion is recursively applied from the root to the leaves of the tree (i.e., top-down).

The JHT contains trees for both representations, with open subtrees and in pure preparation-prolongation form. A python script was used to automatically transform the former into the letter. The script and additional utilities such as for tree traversal and drawing are provided with the treebank.

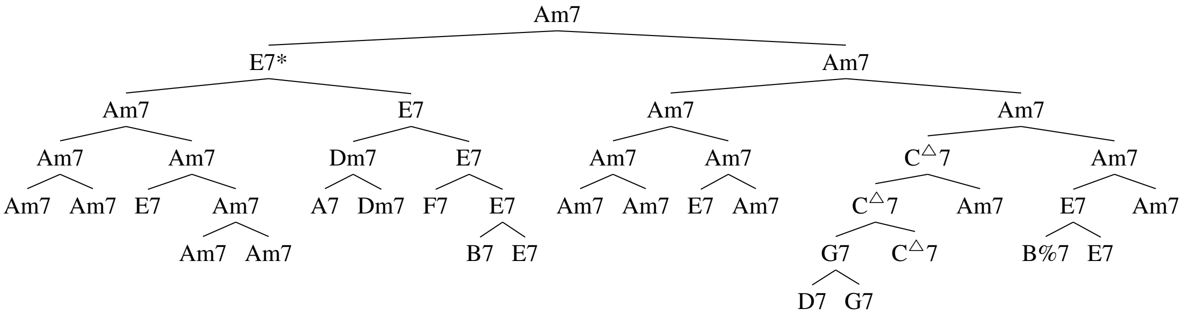


Figure 3: Complete syntax tree of the Jazz standard *Summertime* (turnaround omitted). The top levels of the tree reflect the ABAC form the song using an open constituent.

3. TREE ANNOTATION TOOL

The trees of the JHT are created using a graphical interface implemented as a simple web application, which was developed during the creation of the treebank. The source code of the application is written in ClojureScript (which compiles to JavaScript) and publicly available on GitHub. The application itself is hosted on GitHub pages and can be used independently of this dataset.[5] A screenshot of the application is shown in Figure 1d. The main part of the user interface displays a syntax tree that is represented by a hierarchical button layout. The user interface also contains an input-output section and buttons for creating, deleting, and deselecting tree nodes.

To create a syntax tree, the user inputs a sequence of space-separated strings such as chord symbols. To create an inner node of the tree, the nodes that become the child nodes of the new inner node are selected and combined by pressing a button or a key shortcut. Since the trees are mostly right-headed, the label of the rightmost child is used for the new node by default, but the label of a node can be changed arbitrarily. The output of the application is given as a string representation of the tree in tikz-qtree format as shown in Figure 1c and in JSON format as shown in Figure 1e.[6] Existing trees can be edited by loading them in any of these two formats. Since the application is designed to be agnostic to annotation conventions, it allows arbitrary labels and rule arities.

4. ANNOTATION PROCEDURE

All analyses in the dataset begin from chord sequences drawn from the iRealPro collection of Jazz standards. This collection was created by the user community of the iRealPro app[7] and transferred into kern format by Shanahan et al. [45].[8] We transformed the data into a JSON-like format and occasionally corrected individual chord symbols when we noticed serious differences between the iRealPro data and publicly available *Real Books* (i.e., collections of lead sheets.). Annotations of bass notes and optional chord tones such as ninths and elevenths were excluded from the chord symbols. Chord symbols with a duration of more than one measure were split into multiple chord symbols. 150 Jazz standards were selected for analysis (i) by filtering pieces that are within the scope of the theory of harmonic syntax described in Section 2 and (ii) by preferring shorter pieces. If applicable, turnarounds at the end of a lead sheet were deleted or a final tonic chord not contained in the lead sheet was added. All repetitions were unfolded and codas were appended at the positions indicated in the lead sheet. The selected Jazz standards were initially analyzed by the first author and a student assistant. The analyses were then reviewed by the second and the third author and discussed in the group. To ensure consistent analyses across all 150 Jazz standards, all final tree editing was performed by the first author.

Every hierarchical analysis denotes at least one author's mental representation of the harmonic structure of a Jazz standard. Each analysis is therefore also influenced by other musical features such as harmonic rhythm, phrasing, musical form, and melody. In ambiguous cases, the analyst chose the option that he seemed most important. These choices were necessary, because a single syntax tree can only encode one harmonic function for each chord. For example in the key C major, a C major triad can act as a tonic or as a preparation of a following F major chord. For five particularly ambiguous tunes, we provide alternative analyses in the treebank.

Since the iRealPro lead sheets were created and collected by the community of the application, the chord symbol usage is not fully consistent across the pieces. For instance, a Fm6 chord symbol can denote a tonic chord in F minor over a Dorian scale or a Bb9 chord with omitted root and fifth in the bass. Another example is that fourth-voicings are commonly denoted as suspension chords while actual suspensions of the scale degree V (e.g., suspension of C and E by B and D in a G major triad) are sometimes denoted as chords over the scale degree I (with or without explicitly mentioning the second inversion).

Furthermore, some chords do not have a proper harmonic function, but are better explained as voice-leading connections between two chords. The chords C C#o7 G/D at the beginning of the final 8 measures of *Bill Bailey* are an example of such a voice-leading connection (see Figure 4). Moreover, these final measures are an example of

[5] Link to tree annotation app: `https://dcmlab.github.io/tree-annotation-code/`

[6] `https://www.ctan.org/pkg/tikz-qtree`

[7] `https://irealpro.com/`

[8] The iRealPro dataset is available in kern format at `http://doi.org/10.5281/zenodo.3546040`.

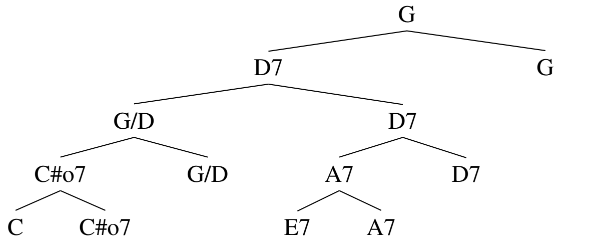


Figure 4: Syntax tree of the final 8 measures of *Bill Bailey* (turnaround omitted).

a common closing pattern. This pattern starts on the scale degree IV in its first measure, then transitions to a suspension of the scale degree V in measure 3, jumps away, and finally approaches the tonic through the cycle of fifths.

5. DATASET SUMMARY

The JHT is provided as a single file in JavaScript Object Notation (JSON) format. For each Jazz standard, this file contains the chord sequence with rhythmical information (measures and beats), metadata about title, composer(s), year of composition, time signature, and key (root & major/minor) as well as the tree analyses. [9]

In addition to the hierarchical analyses, some pieces contain a turnaround annotation represented as an integer. A value of zero means that the Jazz standard ends with a tonic chord. A positive value n means that the lead sheet of the piece ends with a turnaround of length n. For example, the chord sequence of *I love Paris* (in C major) has a turnaround length of $n = 2$, because it ends with the chords Dm7 G7 C6 D%7 G7. A negative turnaround annotation means that the tonic of the piece is not at the end of the piece, but at the beginning. A value of -1 indicates, for example, that the first chord of the chord sequence is the tonic of the piece, like in *Solar*. In rare cases, the tonic is not the first chord but the n-th chord which is represented by a turnaround annotation of $-n$.

The 150 chord sequences analysed in the treebank have an average length of 27.75 and consist of 11697 chords in total with 92 unique chord symbols. The syntax trees consist in total of 3899 binary rule applications with 512 unique rules and 268 open constituents. The average tree height is 7.57.

Further descriptive statistics of the JHT are visualized in Figure 5. The first plot shows that the subset of the analyzed pieces is chosen relatively independently from the year of composition. The second plot shows the bias for short pieces in this subset. The third plot shows that the length of turnarounds, if present, usually ranges between 1 and 3. The two last plots show separately for major and minor keys how often a context-free grammar rule is used in the hierarchical analyses. For these plots, all chord sequences were transposed to C major or to C minor, respectively. Prolongations of the tonic, preparations of the tonic by the fifth scale degree, and preparations of the fifth scale degree by the second are by far the most common rules.

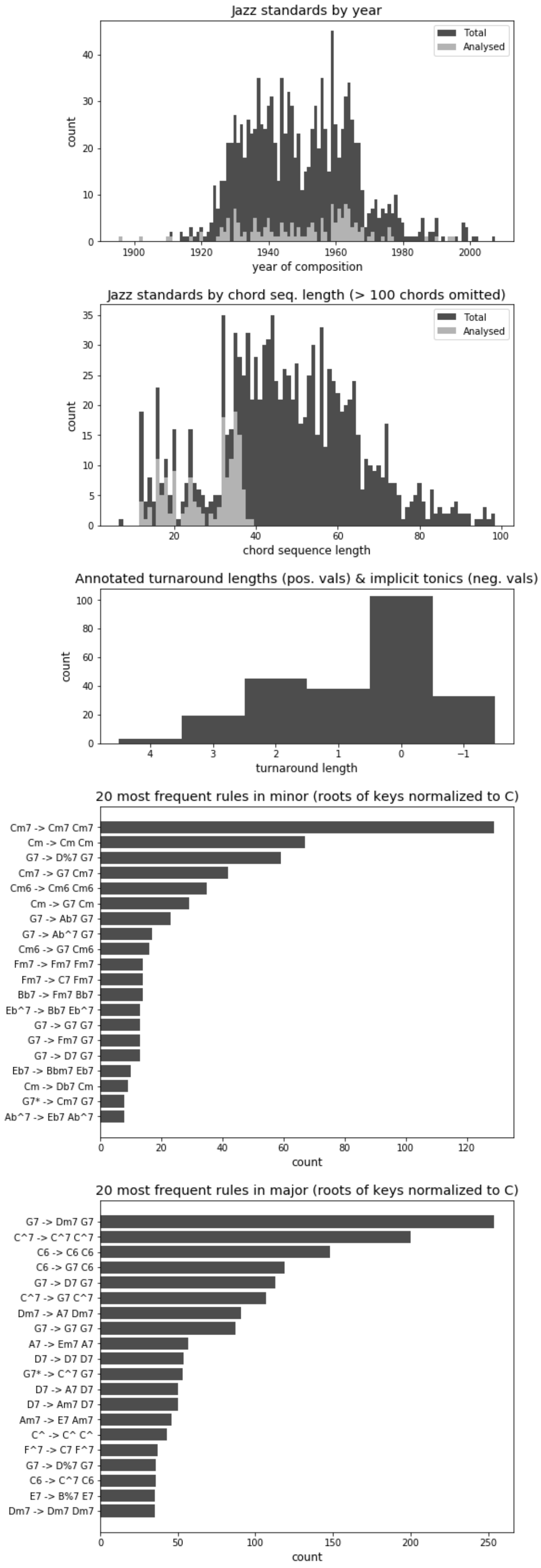


Figure 5: Plots of summary statistics of the tree analyses. See the main text for further explanation.

[9] The metadata was copied from the iRealPro dataset without detailed validity checking. It is provided for convenience.

6. ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 760081 – PMSB. We gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fonds de Recherche du Québec, Société et Culture (FRQSC), and the Canada CIFAR AI Chairs program. We thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology. The authors additionally thank the anonymous referees for their valuable comments and the members of the Digital and Cognitive Musicology Lab (DCML) for fruitful discussions.

7. REFERENCES

[1] Samer Abdallah, Nicolas Gold, and Alan Marsden. Analysing Symbolic Music with Probabilistic Grammars. In David Meredith, editor, *Computational Music Analysis*, pages 157–189. Springer International Publishing, Cham, 2016.

[2] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, Florida, 2011.

[3] Allen Cadwallader and David Gagné. *Analysis of Tonal Music: A Schenkerian Approach*. Oxford University Press, Oxford, 2nd edition, 2007.

[4] William E Caplin. Topics and formal functions the case of the lament. *The Oxford Handbook of Topic Theory*, page 415, 2014.

[5] Tsung-Ping Chen and Li Su. Functional Harmony Recognition of Symbolic Music Data with Multi-task Recurrent Neural Networks. In *ISMIR*, pages 90–97, 2018.

[6] Noam Chomsky. *Aspects of the Theory of Syntax*, volume 11. MIT press, 1965.

[7] W. Bas De Haas, José Pedro Magalhães, Frans Wiering, and Remco C. Veltkamp. Automatic functional harmonic analysis. *Computer Music Journal*, 37(4):37–53, 2013. Publisher: MIT Press.

[8] Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula. Theme And Variation Encodings with Roman Numerals (TAVERN). In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Malaga, Spain, 2015.

[9] Christoph Finkensiep, Richard Widdess, and Martin Rohrmeier. Modelling the Syntax of North Indian Melodies with a Generalized Graph Grammar. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 462–269. ISMIR, 2019.

[10] Francesco Foscarin, Florent Jacquemard, Philippe Rigaux, and Sakai Masahiko. A parse-based framework for coupled rhythm quantization and score structuring. In *Mathematics and Computation in Music*, pages 248–260, Cham, 2019. Springer International Publishing.

[11] Robert Gaizauskas. Investigations into the grammar underlying the Penn Treebank II. *Research Memorandum CS-95-25, Department of Computer Science, Univeristy of Sheffield*, pages 185–189, 1995.

[12] Édouard Gilbert and Darrell Conklin. A probabilistic context-free grammar for melodic reduction. In *Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence*, pages 83–94, 2007.

[13] Mark Gotham and Matthew T. Ireland. Taking Form: A Representation Standard, Conversion Code, and Example Corpus for Recording, Visualizing, and Studying Analyses of Musical Form. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[14] Mark Granroth-Wilding and Mark Steedman. A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43(4):355–374, October 2014.

[15] Ryan Groves. Automatic Melodic Reduction Using a Supervised Probabilistic Context-free Grammar. *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016.

[16] W Bas De Haas, José Pedro Magalhães, and Frans Wiering. Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 295–300, 2012.

[17] Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. Musical Structural Analysis Database based on GTTM. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014.

[18] Daniel Harasim, Timothy J. O'Donnell, and Martin Rohrmeier. Harmonic Syntax in Time: Rhythm Improves Grammatical Models of Harmony. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, 2019.

[19] Daniel Harasim, Martin Rohrmeier, and Timothy J. O'Donnell. A Generalized Parsing Framework for Generative Models of Harmonic Syntax. *19th International Society for Music Information Retrieval Conference*, 2018.

[20] Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gomez. Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations. In *Proceedings of the 6th International Society for Music Information Retrieval Conference*, London, UK, 2005.

[21] Julia Hockenmaier and Mark Steedman. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, 2007.

[22] Jason Katz-Brown, Slav Petrov, Ryan McDonald, Franz Och, David Talbot, Hiroshi Ichikawa, Masakazu Seno, and Hideto Kazawa. Training a parser for machine translation reordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 183–192, USA, 2011. Association for Computational Linguistics.

[23] Phillip B. Kirlin. *A Probabilistic Model of Hierarchical Music Analysis*. PhD thesis, University of Massachusetts Amherst, 2014.

[24] Phillip B. Kirlin and David D. Jensen. Probabilistic Modeling of Hierarchical Music Analysis. In *ISMIR*, pages 393–398, 2011.

[25] Stefan Koelsch, Martin Rohrmeier, Renzo Torrecuso, and Sebastian Jentschke. Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, 110(38):15443–15448, 2013.

[26] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA, 1983.

[27] Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467. Cairo, 2004.

[28] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. 1993.

[29] Andrew McLeod and Mark Steedman. Meter Detection in Symbolic Music Using a Lexicalized PCFG. *Proceedings of the 14th Sound and Music Computing Conference*, 2017.

[30] Andrew McLeod and Mark Steedman. Meter Detection and Alignment of MIDI Performance. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.

[31] Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*, 2017.

[32] Gianluca Micchi, Mark Gotham, and Mathieu Giraud. Not All Roads Lead to Rome: Pitch Representation and Model Architecture for Automatic Harmonic Analysis. *Transactions of the International Society for Music Information Retrieval*, 3(1):42–54, May 2020.

[33] Eita Nakamura, Masatoshi Hamanaka, Keiji Hirata, and Kazuyoshi Yoshii. Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[34] Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Rhythm transcription of MIDI performances based on hierarchical Bayesian modelling of repetition and modification of musical note patterns. In *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016.

[35] Markus Neuwirth, Daniel Harasim, Fabian C. Moss, and Martin Rohrmeier. The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets. *Frontiers in Digital Humanities*, 5, July 2018.

[36] Markus Neuwirth and Martin Rohrmeier. Towards a syntax of the Classical cadence. In *What is a Cadence*, pages 287–338. Leuven University Press, 2015.

[37] Joakim Nivre. Dependency grammar and dependency parsing. *MSI report*, 5133(1959):1–32, 2005.

[38] David Rizo and Alan Marsden. A standard format proposal for hierarchical analyses and representations. In *Proceedings of the 3rd International workshop on Digital Libraries for Musicology*, pages 25–32, 2016.

[39] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.

[40] Martin Rohrmeier. The syntax of jazz harmony: Diatonic tonality, phrase structure, and form. *Music Theory and Analysis*, 7(1):1–63, 2020. Publisher: Leuven University Press.

[41] Martin Rohrmeier and Ian Cross. Tacit tonality-Implicit learning of context-free harmonic structure. In *ESCOM 2009: 7th Triennial Conference of European Society for the Cognitive Sciences of Music*, 2009.

[42] Martin Rohrmeier and Marcus Pearce. Musical syntax I: theoretical perspectives. In *Springer handbook of systematic musicology*, pages 473–486. Springer, 2018.

[43] Anoop Sarkar. Applying co-training methods to statistical parsing. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[44] Heinrich Schenker. *Der Freie Satz. Neue musikalische Theories and Phantasien*. Universal edition edition, 1935.

[45] Daniel Shanahan, Yuri Broze, and Richard Rodgers. A Diachronic Analysis of Harmonic Schemata in Jazz. In *Proceedings of the 12th International Conference on*

Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music, pages 909–917, 2012.

[46] Mark Steedman and Jason Baldridge. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and explicit models of grammar*, pages 181–224, 2011.

Bibliography

[Citing pages are listed after each reference.]

Abdallah, S., Gold, N., and Marsden, A. (2016). Analysing Symbolic Music with Probabilistic Grammars. In Meredith, D., editor, *Computational Music Analysis*, pages 157–189. Springer International Publishing, Cham. [Page 40]

Abdallah, S. A. and Gold, N. E. (2014). Comparing models of symbolic music using probabilistic grammars and probabilistic programming. *Proceedings of ICMC/SMC*, pages 1524–1531. [Pages 2 and 40]

Adger, D. (2003). *Core Syntax: A Minimalist Approach*. Oxford University Press, Oxford. [Page 16]

Agmon, E. (1989). A mathematical model of the diatonic system. *Journal of Music Theory*, 33(1):1–25. [Page 9]

Aho, A. V. (1968). Indexed grammars—an extension of context-free grammars. *Journal of the ACM (JACM)*, 15(4):647–671. [Page 77]

Aldwell, E. and Schachter, C. (2003). *Harmony and Voice Leading*. Thomson Schirmer, New York, second edition. [Page 39]

Anderson, J. R. (1990). *The Adaptive Character of Thought*. Erlbaum, Hillsdale, NJ. [Pages 36 and 37]

Arbib, M. A., editor (2013). *Language, Music, and the Brain: A Mysterious Relationship*. Strüngmann Forum Reports. The MIT Press, Cambridge, MA. [Page 40]

Asano, R. and Boeckx, C. (2015). Syntax in language and music: What is the right level of comparison? *Frontiers in Psychology*, 6. [Page 40]

Baker, J. K. (1979). Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132. [Pages 79 and 88]

Baroni, M. (1999). Musical grammar and the study of cognitive processes of composition. *Musicae Scientiae*, 3(1):3–21. [Page 40]

Baroni, M., Dalmonte, R., and Jacoboni, C. (1992). Theory and analysis of European melody. *Computer Representations and Models in Music*, pages 187–206. [Page 40]

Baroni, M. and Jacoboni, C. (1975). Analysis and generation of Bach's chorale melodies. In *Proceedings of the First International Congress on the Semiotics of Music, Centro Di Iniziativa Culturale*, Pesaro, Italy. [Pages 2 and 40]

Baroni, M. and Jacoboni, C. (1983). Computer generation of melodies: Further proposals. *Computers and the Humanities*, pages 1–18. [Page 40]

Baroni, M., Maguire, S., and Drabkin, W. (1983). The concept of musical grammar. *Music Analysis*, 2(2):175–208. [Page 40]

Berger, J. O. (2013). *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media, New York. [Page 99]

Bernstein, L. (1976). *The Unanswered Question*. Harvard University Press, Cambridge, MA. [Page 42]

Berwick, R. C., Pietroski, P., Yankama, B., and Chomsky, N. (2011). Poverty of the Stimulus Revisited. *Cognitive Science*, 35(7):1207–1242. [Pages 2 and 26]

Bigand, E., Lalitte, P., and Dowling, W. J. (2009). Music and Language: 25 Years After Lerdahl & Jackendoff's GTTM. *Music Perception*, 26(3):185–186. [Pages 2 and 42]

Bigand, E., Parncutt, R., and Lerdahl, F. (1996). Perception of musical tension in short chord sequences: The influence of harmonic function, sensory dissonance, horizontal motion, and musical training. *Perception & Psychophysics*, 58(1):125–141. [Page 43]

Bigand, E. and Poulin-Charronnat, B. (2006). Are we "experienced listeners"? A review of the musical capacities that do not depend on formal musical training. *Cognition*, 100(1):100–130. [Page 24]

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York. [Pages 34, 99, and 104]

Bishop, C. M. (2013). Model-based machine learning. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20120222. [Page 28]

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877. [Pages 93, 95, and 104]

Booth, T. and Thompson, R. (1973). Applying Probability Measures to Abstract Languages. *IEEE Transactions on Computers*, C-22(5):442–450. [Page 73]

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012). Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, Uk. [Page 47]

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2013). Audio Chord Recognition with Recurrent Neural Networks. In *ISMIR*, pages 335–340. Citeseer. [Page 47]

Bratko, I. (1986). *Prolog Programming for Artificial Intelligence*. Addison-Wesley Publishers Limited. [Page 76]

Brighton, H. and Gigerenzer, G. (2008). Bayesian brains and cognitive mechanisms: Harmony or dissonance. In Chater, N. and Oaksford, M., editors, *The Probabilistic Mind: Prospects for Bayesian Cognitive Science*, pages 189—208. Oxford University Press, Oxford. [Page 37]

Brocot, A. (1860). Calcul des rouages par approximation, nouvelle méthode. *Revue Chronométrique*, 6:186–194. [Page 164]

Bubic (2010). Prediction, cognition and the brain. *Frontiers in Human Neuroscience*. [Page 23]

Burgoyne, J. A. (2012). *Stochastic Processes & Database-Driven Musicology*. PhD thesis, McGill University, Montréal, Canada. [Page 29]

Burgoyne, J. A., Wild, J., and Fujinaga, I. (2011). An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, Florida. [Page 52]

Burris, S. and Sankappanavar, H. P. (2012). *A Course in Universal Algebra: The Millenium Edition*. [Page 89]

Cadwallader, A. and Gagné, D. (2007). *Analysis of Tonal Music: A Schenkerian Approach*. Oxford University Press, Oxford, second edition. [Pages 7, 25, and 40]

Calkin, N. and Wilf, H. S. (2000). Recounting the rationals. *The American Mathematical Monthly*, 107(4):360–363. [Page 163]

Callender, C., Quinn, I., and Tymoczko, D. (2008). Generalized voice-leading spaces. *Science*, 320(5874):346–348. [Page 9]

Caplin, W. E. (1998). *Classical Form: A Theory of Formal Functions for the Instrumental Music of Haydn, Mozart, and Beethoven*. Oxford University Press. [Pages 21 and 134]

Caplin, W. E. (2014). Topics and Formal Functions: The Case of the Lament. *The Oxford Handbook of Topic Theory*, page 415. [Page 53]

Carey, N. and Clampitt, D. (1989). Aspects of well-formed scales. *Music Theory Spectrum*, 11(2):187–206. [Page 9]

Chater, N., Tenenbaum, J. B., and Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences*, 10(7):287–291. [Pages 2, 23, and 28]

Chemillier, M. (2004). Toward a formal study of jazz chord sequences generated by Steedman's grammar. *Soft Computing*, 8(9):617–622. [Page 44]

Chen, T.-P. and Su, L. (2018). Functional Harmony Recognition of Symbolic Music Data with Multi-task Recurrent Neural Networks. In *ISMIR*, pages 90–97. [Page 52]

Cherla, S., Weyde, T., Garcez, A., and Pearce, M. (2013). Learning distributed representations for multiple-viewpoint melodic prediction. [Page 46]

Chi, Z. and Geman, S. (1998). Estimation of probabilistic context-free grammars. *Computational linguistics*, 24(2):299–305. [Page 74]

Chomsky, N. (1957). *Syntactic Structures*. Mouton, The Hague. [Page 16]

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA. [Pages 2, 16, 26, 37, 39, 41, 42, and 44]

Chomsky, N. (1971). *Problems of Knowledge and Freedom: The Russell Lectures*. Vintage Books New York. [Page 26]

Chomsky, N. (1975). *Reflections on Language*. Pantheon Books, New York. [Page 26]

Chomsky, N. (1980). *Rules and Representations*. Columbia University Press, New York. [Page 2]

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin, and Use*. Greenwood Publishing Group, London. [Page 2]

Chomsky, N. (1995). *The Minimalist Program*. MIT Press, Cambridge, MA. [Page 16]

Clark, A. (2013a). Learning trees from strings: A strong learning algorithm for some context-free grammars. *The Journal of Machine Learning Research*, 14(1):3537–3559. [Page 149]

Clark, A. (2013b). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204. [Page 37]

Clarke, D. (2017). North Indian Classical Music and Lerdahl and Jackendoff's Generative Theory–a Mutual Regard. *Music Theory Online*, 23(3). [Page 41]

Clough, J. and Douthett, J. (1991). Maximally Even Sets. *Journal of Music Theory*, 35(1/2):93. [Page 9]

Clough, J. and Myerson, G. (1985). Variety and multiplicity in diatonic systems. *Journal of Music Theory*, 29(2):249–270. [Page 9]

Cohn, R. (1996). Maximally smooth cycles, hexatonic systems, and the analysis of late-romantic triadic progressions. *Music Analysis*, 15(1):9—40. [Page 9]

Cohn, R. (1997). Neo-riemannian operations, parsimonious trichords, and their" tonnetz" representations. *Journal of Music Theory*, 41(1):1–66. [Pages 9, 22, and 44]

Cohn, R. (1998). Introduction to neo-riemannian theory: A survey and a historical perspective. *Journal of Music Theory*, 42(2):167—180. [Page 9]

Cohn, T., Blunsom, P., and Goldwater, S. (2010). Inducing tree-substitution grammars. *The Journal of Machine Learning Research*, 11:3053–3096. [Page 149]

Colmerauer, A. (1978). Metamorphosis grammars. In *Natural Language Communication with Computers*, pages 133–188. Springer. [Page 76]

Cone, E. T. (1968). *Musical Form and Musical Performance*. Norton, New York. [Pages 21 and 42]

Conklin, D. and Cleary, J. G. (1988). Modelling and generating music using multiple viewpoints. In *Proeedings of the 1st Workshop AI Music*, pages 125–137, Menlo Park. University of Calgary. [Page 46]

Conklin, D. and Witten, I. H. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73. [Page 46]

Cooper, G. and Meyer, L. B. (1960). *The Rhythmic Structure of Music*. University of Chicago Press, Chicago. [Pages 21 and 42]

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American journal of physics*, 14(1). [Page 29]

Cox, R. T. (1961). *The Algebra of Probable Inference*. Johns Hopkins University Press, Baltimore, MD. [Page 29]

Crain, S. (1991). Language acquisition in the absence of experience. *Behavioral and Brain Sciences*, 14(4):597–612. [Page 26]

Crain, S. and Nakayama, M. (1987). Structure dependence in grammar formation. *Language*, pages 522–543. [Page 26]

Cross, I. (1998). Music analysis and music perception. *Music Analysis*, 17(1):3–20. [Pages 41 and 180]

De Clercq, T. and Temperley, D. (2011). A corpus analysis of rock harmony. *Popular Music*, pages 47–70. [Page 46]

De Haas, W. B. (2012). *Music Information Retrieval Based on Tonal Harmony*. PhD thesis, Utrecht University, Utrecht. [Pages 2 and 39]

De Haas, W. B., Magalhães, J. P., and Wiering, F. (2012). Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. In *ISMIR*, pages 295–300. Citeseer. [Pages 2 and 39]

De Haas, W. B., Rohrmeier, M., Veltkamp, R. C., and Wiering, F. (2009). Modeling harmonic similarity using a generative grammar of tonal harmony. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*. [Pages 2 and 39]

Déguernel, K., Vincent, E., Nika, J., Assayag, G., and Smaïli, K. (2019). Learning of Hierarchical Temporal Structures for Guided Improvisation. *Computer Music Journal*, 43(2). [Page 150]

Deliege, I. (1987). Grouping conditions in listening to music: An approach to Lerdahl & Jackendoff's grouping preference rules. *Music perception*, 4(4):325–359. [Page 42]

Devaney, J., Arthur, C., Condit-Schultz, N., and Nisula, K. (2015). Theme And Variation Encodings with Roman Numerals (TAVERN). In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Malaga, Spain. [Page 52]

Dienes, Z. and Perner, J. (1999). A theory of implicit and explicit knowledge. *Behavioral and brain sciences*, 22(5):735–808. [Page 24]

Diergarten, F. and Neuwirth, M. (2019). *Formenlehre: Ein Lese-Und Arbeitsbuch Zur Instrumentalmusik Des 18. Und 19. Jahrhunderts*. Laaber. [Page 20]

Dolan, S. (2013). Fun with semirings: A functional pearl on the abuse of linear algebra. In *Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming*, pages 101–110. [Pages 87 and 88]

Domínguez, M., Clampitt, D., and Noll, T. (2007). WF scales, ME sets, and Christoffel words. In *International Conference on Mathematics and Computation in Music*, pages 477–488. Springer. [Page 9]

Douthett, J. and Steinbach, P. (1998). Parsimonious graphs: A study in parsimony, contextual transformations, and modes of limited transposition. *Journal of Music Theory*, pages 241–263. [Pages 9 and 22]

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94—102. [Pages 38 and 79]

Egermann, H., Pearce, M. T., Wiggins, G. A., and McAdams, S. (2013). Probabilistic models of expectation violation predict psychophysiological emotional responses to live concert music. *Cognitive, Affective, & Behavioral Neuroscience*, 13(3):533–553. [Page 13]

Euler, L. (1739). *Tentamen Novae Theoriae Musicae Ex Certissimis Harmoniae Principiis Dilucide Expositae*. Ex Typographia Academiae Scientiarum, St. Petersburg. [Page 44]

Everett, W. (2004). Making sense of rock's tonal systems. *Music Theory Online*, 10(4). [Page 41]

Fine, T. L. (1973). *Theories of Probability: An Examination of Foundations*. Academic Press, New York and London. [Pages 29 and 30]

Finkensiep, C., Neuwirth, M., and Rohrmeier, M. (2018). Generalized skipgrams for pattern discovery in polyphonic streams. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 547–553, Paris, France. [Page 46]

Finkensiep, C., Widdess, R., and Rohrmeier, M. (2019). Modelling the Syntax of North Indian Melodies with a Generalized Graph Grammar. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, pages 462–269. ISMIR. [Page 41]

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345. [Page 87]

Fodor, J. A. (1981). *Representations: Philosophical Essays on the Foundations of Cognitive Science*. MIT Press, Cambridge, MA. [Page 23]

Forte, A. (1959). Schenker's conception of musical structure. *Journal of Music Theory*, 3(1):1–30. [Page 40]

Forte, A. (1995). *The American Popular Ballad of the Golden Era 1924–1950*. Oxford University Press, New York. [Page 21]

Frankel, R. E., Rosenschein, S. J., and Smoliar, S. W. (1976). A LISP-based system for the study of Schenkerian analysis. *Computers and the Humanities*, pages 21–32. [Page 41]

Frankel, R. E., Rosenschein, S. J., and Smoliar, S. W. (1978). Schenker's theory of tonal music—its explication through computational processes. *International Journal of Man-Machine Studies*, 10(2):121–138. [Page 41]

Frankland, B. W. and Cohen, A. J. (2004). Parsing of melody: Quantification and testing of the local grouping rules of Lerdahl and Jackendoff's A Generative Theory of Tonal Music. *Music Perception*, 21(4):499–543. [Page 42]

Frieler, K. (2014). Pattern usage in monophonic jazz solos. In *International Jazzomat Research Workshop*, Weimar, Germany. [Page 46]

Frieler, K. (2019). Constructing Jazz Lines Taxonomy, Vocabulary, Grammar. In Pfleiderer, M. and Zaddach, W.-G., editors, *Jazzforschung Heute: Themen, Methoden, Perspektiven*, pages 103–132. EDITION EMVAS, Berlin. [Page 46]

Frieler, K. (2020). Miles Vs. Trane. Computational and Statistical Comparison of the Improvisatory Styles of Miles Davis and John Coltrane. *Jazz Perspectives*, 12(1):123–145. [Page 46]

Friston, K. (2009). The free-energy principle: A rough guide to the brain? *Trends in Cognitive Sciences*, 13(7):293–301. [Page 37]

Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature reviews neuroscience*, 11(2):127–138. [Page 37]

Gaizauskas, R. (1995). Investigations into the grammar underlying the Penn Treebank II. *Research Memorandum CS-95-25, Department of Computer Science, Univeristy of Sheffield*, pages 185–189. [Page 52]

Giblin, I. (2008). *Music and the Generative Enterprise: Situating a Generative Theory of Tonal Music in the Cognitive Sciences*. PhD thesis, University of New South Wales, Sydney. [Pages 2 and 42]

Gilbert, É. and Conklin, D. (2007). A probabilistic context-free grammar for melodic reduction. In *Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence*, pages 83–94. [Pages 2, 7, and 40]

Givan, B. (2010). Swing Improvisation: A Schenkerian Perspective. *Theory and Practice*, pages 25–56. [Page 41]

Golland, D., DeNero, J., and Uszkoreit, J. (2012). A feature-rich constituent context model for grammar induction. [Page 149]

Gollin, E. (2006). Some further notes on the history of the Tonnetz. *Theoria: Historical Aspects of Music Theory*, 13:99–111. [Page 44]

Gollin, E. H. (2000). *Representations of Space and Conceptions of Distance in Transformational Music Theories*. PhD thesis, Harvard University. [Page 9]

Gomez, R. L. and Gerken, L. (1999). Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge. *Cognition*, 70(2):109–135. [Page 27]

Goodman, J. (1998). *Parsing Inside-Out*. PhD thesis, Harvard University, Cambridge, Massachusetts. [Pages 3 and 79]

Goodman, J. (1999). Semiring Parsing. *Computational Linguistics*, 25(4):34. [Pages 3, 79, 84, and 86]

Gopnik, A. and Tenenbaum, J. B. (2007). Bayesian networks, Bayesian learning and cognitive development. *Developmental Science*, 10(3):281–287. [Page 37]

Gotham, M. and Ireland, M. T. (2019). Taking Form: A Representation Standard, Conversion Code, and Example Corpus for Recording, Visualizing, and Studying Analyses of Musical Form. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands. [Page 52]

Granroth-Wilding, M. (2013). *Harmonic Analysis of Music Using Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh. [Page 45]

Granroth-Wilding, M. and Steedman, M. (2014). A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43(4):355–374. [Pages 2, 39, 45, 52, and 113]

Griffiths, T. L., Kemp, C., and Tenenbaum, J. B. (2008). Bayesian models of cognition. In Sun, R., editor, *The Cambridge Handbook of Computational Cognitive Modeling*. Cambridge University Press, Cambridge. [Pages 2, 26, 28, and 33]

Groves, R. (2016). Automatic Melodic Reduction Using a Supervised Probabilistic Context-free Grammar. *Proceedings of the 17th International Society for Music Information Retrieval Conference*. [Page 43]

Hadjeres, G., Pachet, F., and Nielsen, F. (2017). Deepbach: A steerable model for bach chorales generation. In *International Conference on Machine Learning*, pages 1362–1371. [Page 47]

Hájek, A. (2019). Interpretations of probability. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition. [Page 28]

Halsey, L. G., Curran-Everett, D., Vowler, S. L., and Drummond, G. B. (2015). The fickle P value generates irreproducible results. *Nature methods*, 12(3):179–185. [Page 125]

Hamanaka, M., Hirata, K., and Tojo, S. (2005). ATTA: Automatic Time-Span Tree Analyzer Based on Extended GTTM. In *ISMIR*, volume 5, pages 358–365. [Page 43]

Hamanaka, M., Hirata, K., and Tojo, S. (2006). Implementing "A generative theory of tonal music". *Journal of New Music Research*, 35(4):249–277. [Page 43]

Hamanaka, M., Hirata, K., and Tojo, S. (2007a). ATTA: Implementing GTTM on a Computer. In *ISMIR*, pages 285–286. [Page 43]

Hamanaka, M., Hirata, K., and Tojo, S. (2007b). FATTA: Full automatic time-span tree analyzer. In *ICMC*, volume 1, pages 153–156. Citeseer. [Page 43]

Hamanaka, M., Hirata, K., and Tojo, S. (2014). Musical Structural Analysis Database based on GTTM. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan. [Pages 43 and 52]

Hamanaka, M., Hirata, K., and Tojo, S. (2015). σGTTM III: Learning-Based Time-Span Tree Generator Based on PCFG. In *International Symposium on Computer Music Multidisciplinary Research*, pages 387–404. Springer. [Page 43]

Hamanaka, M., Hirata, K., and Tojo, S. (2016a). deepGTTM-I&II: Local Boundary and Metrical Structure Analyzer Based on Deep Learning Technique. In *International Symposium on Computer Music Multidisciplinary Research*, pages 3–21. Springer. [Page 43]

Hamanaka, M., Hirata, K., and Tojo, S. (2016b). Implementing methods for analysing music based on lerdahl and jackendoff's generative theory of tonal music. In *Computational Music Analysis*, pages 221–249. Springer. [Page 43]

Hamanaka, M., Isono, Y., Hirata, K., and Tojo, S. (2020). Web-based Time-span Tree Editor and Analysis Database. In *Proceedings of the 17th Sound and Music Computing Conference*, Torino. [Page 43]

Hamanaka, M. and Tojo, S. (2009). Interactive Gttm Analyzer. In *ISMIR*, pages 291–296. [Page 43]

Harasim, D., Finkensiep, C., Ericson, P., O'Donnell, T. J., and Rohrmeier, M. (2020a). The Jazz Harmony Treebank. In *Proceedings of the 21th International Society for Music Information Retrieval Conference*, Montréal, Canada. [Page 4]

Harasim, D., Noll, T., and Rohrmeier, M. (2019a). Distant Neighbors and Interscalar Contiguities. In Montiel, M., Gomez-Martin, F., and Agustín-Aquino, O. A., editors, *Mathematics and Computation in Music*, volume 11502, pages 172–184. Springer International Publishing, Cham. [Page 9]

Harasim, D., O'Donnell, T. J., and Rohrmeier, M. (2019b). Harmonic Syntax in Time: Rhythm Improves Grammatical Models of Harmony. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft. [Pages 4 and 179]

Harasim, D., Rohrmeier, M., and O'Donnell, T. J. (2018). A Generalized Parsing Framework for Generative Models of Harmonic Syntax. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris. [Pages 4, 14, and 84]

Harasim, D., Schmidt, S. E., and Rohrmeier, M. (2016). Bridging scale theory and geometrical approaches to harmony: The voice-leading duality between complementary chords. *Journal of Mathematics and Music*, 10(3):193–209. [Pages 9 and 22]

Harasim, D., Schmidt, S. E., and Rohrmeier, M. (2020b). Axiomatic scale theory. *Journal of Mathematics and Music*. [Page 9]

Harte, C., Sandler, M., Abdallah, S., and Gomez, E. (2005). Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations. In *Proceedings of the 6th International Society for Music Information Retrieval Conference*, London. [Page 52]

Hauser, M. D., Chomsky, N., and Fitch, W. T. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298(5598):1569–1579. [Pages 2 and 26]

Have, C. T. (2009). Stochastic definite clause grammars. In *Proceedings of the International Conference RANLP-2009*, pages 139–143. [Page 76]

Heyer, D. J. (2012). Applying Schenkerian Theory to Mainstream Jazz: A Justification for an Orthodox Approach. *Music Theory Online*, 18(3). [Pages 7 and 41]

Hierarchy (2020). *Oxford English Dictionary Online*. Oxford University Press. [Page 10]

Ho, J., Tumkaya, T., Aryal, S., Choi, H., and Claridge-Chang, A. (2019). Moving beyond P values: Data analysis with estimation graphics. *Nature methods*, 16(7):565–566. [Page 125]

Hockenmaier, J. and Steedman, M. (2007). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396. [Page 52]

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347. [Pages 106, 151, and 172]

Hotz, G. and Pitsch, G. (1996). On parsing coupled-context-free languages. *Theoretical Computer Science*, 161:205–233. [Page 75]

Huron, D. (2006). *Sweet Anticipation: Music and the Psychology of Expectation.* MIT Press, Cambridge, MA. [Pages 13 and 24]

Huron, D. (2012). Two challenges in cognitive musicology. *Topics in cognitive science*, 4(4):678–684. [Page 24]

Huron, D. (2016). *Voice Leading: The Science behind a Musical Art.* MIT Press. [Page 22]

Ihringer, T. and Gumm, H.-P. (2003). *Allgemeine Algebra: Mit Einem Anhang Über Universelle Coalgebra von HP Gumm.* Heldermann. [Page 89]

Jackendoff, R. (1991). Musical parsing and musical affect. *Music Perception*, 9(2):199–229. [Page 13]

Jackendoff, R. and Lerdahl, F. (2006). The capacity for music: What is it, and what's special about it? *Cognition*, 100(1):33–72. [Pages 3, 13, 22, 23, 26, and 42]

Jackendoff, R. and Pinker, S. (2005). The nature of the language faculty and its implications for evolution of language (Reply to Fitch, Hauser, and Chomsky). page 15. [Pages 2 and 26]

Jaynes, E. T. (1986). Bayesian Methods: General Background. In Justice, J. H., editor, *Maximum-Entropy and Bayesian Methods in Applied Statistics.* Cambridge University Press, Cambridge. [Page 29]

Jaynes, E. T. (2003). *Probability Theory: The Logic of Science.* Cambridge University Press. [Pages 28 and 29]

Jeffreys, H. (1939). *Theory of Probability.* Oxford University Press, Oxford. [Page 28]

Johnson, M., Griffiths, T. L., and Goldwater, S. (2007a). Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems*, pages 641–648. [Page 76]

Johnson, M., Griffiths, T. L., and Goldwater, S. (2007b). Bayesian inference for pcfgs via markov chain monte carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146. [Pages 34 and 149]

Johnson-Laird, P. N. (1991). Jazz improvisation: A theory at the computational level. In Howell, P., West, R., and Cross, I., editors, *Representing Musical Structure*, pages 291–325. Academic Press, London, UK. [Page 38]

Jonaitis, E. M. and Saffran, J. R. (2009). Learning Harmony: The Role of Serial Statistics. *Cognitive Science*, 33(5):951–968. [Page 24]

Joshi, A. K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Dowty, D., Karttunen, L., and Zwicky, A., editors, *Natural Language Processing–Theoretical, Computational, and Psychological Perspectives*. Cambridge University Press, New York, USA. [Page 76]

Joshi, A. K., Levy, L. S., and Takahashi, M. (1975). Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1):136–163. [Page 76]

Joshi, A. K. and Schabes, Y. (1997). Tree-adjoining grammars. In *Handbook of Formal Languages*, pages 69–123. Springer. [Page 76]

Joshi, A. K., Shanker, K. V., and Weir, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In Sells, P., Shieber, S., and Wasow, T., editors, *Foundational Issues in Natural Language Processing*. MIT Press, Cambridge, MA. [Page 76]

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA. [Page 80]

Juslin, P. N. and Vastfjall, D. (2008). Emotional responses to music: The need to consider underlying mechanisms. *Behavioral and brain sciences*, 31(5):559. [Page 13]

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the american statistical association*, 90(430):773–795. [Pages 125 and 131]

Katz, J. (2017). Harmonic Syntax of the Twelve-Bar Blues Form: A Corpus Study. *Music Perception: An Interdisciplinary Journal*, 35(2):165–192. [Page 44]

Katz, J. and Pesetsky, D. (2011). The Identity Thesis for Language and Music. [Pages 40 and 182]

Katz-Brown, J., Petrov, S., McDonald, R., Och, F., Talbot, D., Ichikawa, H., Seno, M., and Kazawa, H. (2011). Training a parser for machine translation reordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 183–192, USA. Association for Computational Linguistics. [Page 52]

Keiler, A. (1978). Bernstein's "The Unanswered Question" and the Problem of Musical Competence. *Musical Quarterly*, 62(2):195–222. [Page 41]

Kemp, C. and Tenenbaum, J. B. (2008). The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692. [Page 26]

Kenny, B. J. and Gellrich, M. (2002). Improvisation. In Parncutt, R. and McPherson, G. E., editors, *The Science and Psychology of Music Performance: Creative Strategies for Teaching and Learning*, pages 163–180. Oxford University Press, Oxford. [Page 1]

Kernfeld, B. (2006). *The Story of Fake Books: Bootlegging Songs to Musicians*, volume 53. Scarecrow Press. [Page 9]

Keynes, J. M. (1929). *A Treatise on Probability*. London. [Page 28]

Khadkevich, M. and Omologo, M. (2009). Use of Hidden Markov Models and Factored Language Models for Automatic Chord Recognition. In *ISMIR*, pages 561–566. [Page 46]

Kim, Y., Dyer, C., and Rush, A. (2019). Compound Probabilistic Context-Free Grammars for Grammar Induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics. [Pages 149, 168, and 179]

Kirlin, P. B. (2014). *A Probabilistic Model of Hierarchical Music Analysis*. PhD thesis, University of Massachusetts Amherst. [Pages 41 and 52]

Kirlin, P. B. and Jensen, D. D. (2011). Probabilistic Modeling of Hierarchical Music Analysis. In *ISMIR*, pages 393–398. [Page 41]

Kirlin, P. B. and Jensen, D. D. (2015). Using Supervised Learning to Uncover Deep Musical Structure. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. [Page 41]

Kirlin, P. B. and Thomas, D. L. (2015). Extending a Model of Monophonic Hierarchical Music Analysis to Homophony. In *Proceedings of the 16th International Society for Music Information RetrievalConference*. [Page 41]

Kirlin, P. B. and Utgoff, P. E. (2008). A Framework for Automated Schenkerian Analysis. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 363–368. [Page 41]

Klein, D. and Manning, C. D. (2002). A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 128–135. Association for Computational Linguistics. [Page 149]

Klein, D. and Manning, C. D. (2004). Parsing and Hypergraphs. In Ide, N., Véronis, J., Baayen, H., Church, K. W., Klavans, J., Barnard, D. T., Tufis, D., Llisterri, J., Johansson, S., Mariani, J., Bunt, H., Carroll, J., and Satta, G., editors, *New Developments in Parsing Technology*, volume 23, pages 351–372. Springer Netherlands, Dordrecht. [Page 79]

Knill, D. C. and Pouget, A. (2004). The Bayesian brain: The role of uncertainty in neural coding and computation. *Trends in Neurosciences*, 27(12):712–719. [Page 37]

Kolmogorov, A. (1933). *Foundations of the Theory of Probability*. Chelsea Publishing Company, New York, USA, english translation from 1950 edition. [Page 29]

Koops, H. V., de Haas, W. B., Burgoyne, J. A., Bransen, J., Kent-Muller, A., and Volk, A. (2019). Annotator subjectivity in harmony annotations of popular music. *Journal of New Music Research*, pages 1–21. [Page 179]

Korzeniowski, F., Sears, D. R., and Widmer, G. (2018). A large-scale study of language models for chord prediction. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 91–95. IEEE. [Page 47]

Korzeniowski, F. and Widmer, G. (2016). A fully convolutional deep auditory model for musical chord recognition. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE. [Page 47]

Korzeniowski, F. and Widmer, G. (2018). Improved Chord recognition by combining duration and harmonic language models. In *ISMIR*. [Page 47]

Kostka, S. and Payne, D. (1984). *Tonal Harmony with an Introduction to 20th-Century Music*. McGraw-Hill, New York. [Page 14]

Krumhansl, C. L. (1983). Perceptual structures for tonal music. *Music Perception*, 1(1):28–62. [Page 43]

Krumhansl, C. L. (1990). *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York. [Page 43]

Krumhansl, C. L. (1996). A perceptual analysis of Mozart's Piano Sonata K. 282: Segmentation, tension, and musical ideas. *Music perception*, 13(3):401–432. [Page 43]

Kuhn, G. and Dienes, Z. (2005). Implicit learning of nonlocal musical rules: Implicitly learning more than chunks. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(6):1417. [Page 24]

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86. [Page 104]

Kurihara, K. and Sato, T. (2004). An application of the variational Bayesian approach to probabilistic context-free grammars. In *IJCNLP-04 Workshop beyond Shallow Analyses*. [Pages 3, 34, and 93]

Kurihara, K. and Sato, T. (2006). Variational Bayesian Grammar Induction for Natural Language. In Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., and Tomita, E., editors, *Grammatical Inference: Algorithms and Applications*, volume 4201, pages 84–96. Springer Berlin Heidelberg, Berlin, Heidelberg. [Page 34]

Lange, M. and Leiß, H. (2009). To CNF or not to CNF? An Efficient Yet Presentable Version of the CYK Algorithm. *Informatica didactica*, 8. [Page 84]

Large, E. W. and Palmer, C. (2002). Perceiving temporal regularity in music. *Cognitive science*, 26(1):1–37. [Page 42]

Lari, K. and Young, S. J. (1990). The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer speech & language*, 4(1):35–56. [Pages 79, 88, and 149]

Larson, S. (1998). Schenkerian analysis of modern jazz: Questions about method. *Music Theory Spectrum*, 20(2):209–241. [Pages 7 and 41]

Larson, S. (2002). Musical forces, melodic expectation, and jazz melody. *Music Perception*, 19(3):351–385. [Page 41]

Larson, S. (2009). *Analyzing Jazz: A Schenkerian Approach*. Pendragon Press, New York. [Page 41]

Lavine, M. and Schervish, M. J. (1999). Bayes factors: What they are and what they are not. *The American Statistician*, 53(2):119–122. [Page 125]

Lee, K. and Slaney, M. (2006). Automatic Chord Recognition from Audio Using a HMM with Supervised Learning. In *ISMIR*, pages 133–137. [Page 46]

Lee, K. and Slaney, M. (2008). Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on audio, speech, and language processing*, 16(2):291–301. [Page 46]

Legate, J. A. and Yang, C. D. (2002). Empirical re-assessment of stimulus poverty arguments. *The Linguistic Review*, 19(1-2):151–162. [Page 26]

Lehmann, D. J. (1977). Algebraic structures for transitive closure. *Theoretical Computer Science*, 4(1):59–76. [Page 87]

Lerdahl, F. (1988). Tonal Pitch Space. *Music Perception*, 5(3):315–350. [Page 43]

Lerdahl, F. (1996). Calculating tonal tension. *Music Perception*, 13(3):319–363. [Page 43]

Lerdahl, F. (2001). *Tonal Pitch Space*. Oxford University Press. [Pages 13 and 43]

Lerdahl, F. (2009). Genesis and Architecture of the GTTM Project. *Music Perception: An Interdisciplinary Journal*, 26(3):187–194. [Pages 42 and 43]

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA. [Pages 2, 7, 13, 17, 21, 25, 42, and 52]

Lerdahl, F. and Krumhansl, C. L. (2007). Modeling tonal tension. *Music perception*, 24(4):329–366. [Page 43]

Levine, M. (1990). *The Jazz Piano Book*. Sher Music, Petaluma, CA. [Page 9]

Levine, M. (1995). *The Jazz Theory Book*. Sher Music, Petaluma, CA. [Page 9]

Lewis, J. D. and Elman, J. L. (2001). Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. In *Proceedings of the 26th Annual Boston University Conference on Language Development*, volume 1, pages 359–370. Citeseer. [Pages 2 and 27]

London, J. (2012). *Hearing in Time: Psychological Aspects of Musical Meter.* Oxford University Press. [Page 21]

Longuet-Higgins, H. C. (1962). Letter to a musical friend. *Music review*, 23:244–248. [Page 44]

Longuet-Higgins, H. C. (1976). Perception of melodies. *Nature*, 263(5579):646–653. [Page 44]

Longuet-Higgins, H. C. (1979). The Perception of Music. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 205(1160):307–322. [Pages 1 and 44]

Longuet-Higgins, H. C. and Lee, C. S. (1984). The Rhythmic Interpretation of Monophonic Music. *Music Perception*, 1(4):424–441. [Page 44]

Longuet-Higgins, H. C. and Lisle, E. R. (1989). Modelling musical cognition. *Contemporary Music Review*, 3(1):15–27. [Page 44]

López, D., Sempere, J. M., and García, P. (2004). Inference of reversible tree languages. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1658–1665. [Page 149]

Loui, P. (2012). Learning and Liking of Melody and Harmony: Further Studies in Artificial Grammar Learning. *Topics in Cognitive Science*, 4(4):554–567. [Page 24]

Loui, P., Wu, E. H., Wessel, D. L., and Knight, R. T. (2009). A Generalized Mechanism for Perception of Pitch Patterns. *Journal of Neuroscience*, 29(2):454–459. [Page 24]

Love, S. (2012). An approach to phrase rhythm in jazz. *Journal of Jazz Studies*, 8(1):4–32. [Page 21]

Lovell, J. (2007). The Story of Fake Books: Bootlegging Songs to Musicians (book review). *American Music*, 25(3):370–373. [Page 9]

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, volume 27, pages 466–467. Cairo. [Page 52]

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, Cambridge. [Pages 34, 119, and 169]

MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk. Third Edition.* Lawrence Erlbaum Associates, Mahwah, NJ. [Page 27]

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.* MIT press, Cambridge, MA. [Page 16]

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. [Page 52]

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* New York, NY. [Pages 25 and 36]

Marsden, A. (2001). Representing melodic patterns as networks of elaborations. *Computers and the Humanities*, 35(1):37–54. [Page 41]

Marsden, A. (2005). Generative structural representation of tonal music. *Journal of New Music Research*, 34(4):409–428. [Page 41]

Marsden, A. (2007). Automatic derivation of musical structure: A tool for research on Schenkerian analysis. In *ISMIR*, Vienna. [Page 41]

Marsden, A. (2010). Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, 39(3):269–289. [Page 41]

Martin, H. (1996). *Charlie Parker and Thematic Improvisation.* Scarecrow Press, Lanham, MD. [Page 41]

Martin, H. (2011a). More Than Just Guide Tones: Steve Larson's Analyzing Jazz—A Schenkerian Approach. *Journal of Jazz Studies*, 7(1):121. [Page 41]

Martin, H. (2011b). Schenker and the Tonal Jazz Repertory. *Dutch Journal of Music Theory*, 16(1):1–20. [Page 41]

Mavromatis, P. and Brown, M. (2004). Parsing context-free grammars for music: A computational model of Schenkerian analysis. In *Proceedings of the 8th International Conference on Music Perception & Cognition*, pages 414–415. [Page 41]

McAdams, S. (1989). Psychological constraints on form-bearing dimensions in music. *Contemporary Music Review*, 4(1):181–198. [Page 42]

McCormack, J. (1996). Grammar based music composition. *Complex systems*, 96:321–336. [Pages 2 and 40]

McLeod, A. and Steedman, M. (2017). Meter Detection in Symbolic Music Using a Lexicalized PCFG. *Proceedings of the 14th Sound and Music Computing Conference*, page 7. [Page 179]

Melis, G., Dyer, C., and Blunsom, P. (2017). On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589.* [Page 52]

Meyer, L. B. (1956). *Emotion and Meaning in Music.* University of Chicago Press, Chicago. [Page 13]

Meyer, L. B. (1967). *Music, the Arts, and Ideas: Patterns and Predictions in Twentieth-Century Culture.* University of Chicago Press, Chicago. [Page 13]

Meyer, L. B. (1973). *Explaining Music.* University of California Press, Berkley, CA. [Page 13]

Meyer, L. B. (1989). *Style and Music: Theory, History, and Ideology*. University of Pennsylvania Press, Philadelphia. [Page 16]

Micchi, G., Gotham, M., and Giraud, M. (2020). Not All Roads Lead to Rome: Pitch Representation and Model Architecture for Automatic Harmonic Analysis. *Transactions of the International Society for Music Information Retrieval*, 3(1):42–54. [Page 52]

Michaelis, J. (1998). Derivational minimalism is mildly context–sensitive. In *International Conference on Logical Aspects of Computational Linguistics*, pages 179–198. Springer. [Page 76]

Moss, F. C. (2019). *Transitions of Tonality: A Model-Based Corpus Study*. PhD thesis, École polytechnique fédérale de Lausanne, Lausanne, Switzerland. [Pages 44 and 46]

Moss, F. C., Neuwirth, M., Harasim, D., and Rohrmeier, M. (2019). Statistical characteristics of tonal harmony: A corpus study of Beethoven's string quartets. *PLOS ONE*, 14(6). [Pages 46 and 52]

Moss, F. C., Souza, W. F., and Rohrmeier, M. (2020). Harmony and form in Brazilian Choro: A corpus-driven approach to musical style analysis. *Journal of New Music Research*. [Pages 21 and 52]

Narmour, E. (1990). *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press, Chicago. [Page 13]

Narmour, E. (1991). The Top-down and Bottom-up Systems of Musical Implication: Building on Meyer's Theory of Emotional Syntax. page 26. [Page 13]

Narmour, E. (1992). *The Analysis And Cognition Of Melodic Complextiy: The Implication-Realization Model*. University of Chicago Press, Chicago. [Page 13]

Neuwirth, M., Harasim, D., Moss, F. C., and Rohrmeier, M. (2018). The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets. *Frontiers in Digital Humanities*, 5. [Page 52]

Neuwirth, M. and Rohrmeier, M. (2016). Wie wissenschaftlich muss Musiktheorie sein?. Chancen und Herausforderungen musikalischer Korpusforschung. *Zeitschrift der Gesellschaft für Musiktheorie [Journal of the German-Speaking Society of Music Theory]*, 13(2):171–193. [Pages 41 and 180]

Ogura, Y., Ohmura, H., Uehara, Y., Tojo, S., and Katsurada, K. (2020). Expectation-based Parsing for Jazz Chord Sequences. In *Proceedings of the 17th Sound and Music Computing Conference*, Torino. [Page 38]

Pachet, F. (1997). Computer Analysis of Jazz Chord Sequences: Is Solar a Blues? In Miranda, E., editor, *Readings in Music and Artificial Intelligence*. Harwood Academic Publishers. [Page 44]

Palmer, C. and Krumhansl, C. L. (1990). Mental representations for musical meter. *Journal of Experimental Psychology: Human Perception and Performance*, 16(4):728. [Page 42]

Pearce, M., Conklin, D., and Wiggins, G. (2005). Methods for Combining Statistical Models of Music. In Wiil, U. K., editor, *Computer Music Modeling and Retrieval*, volume 3310, pages 295–312. Springer Berlin Heidelberg, Berlin, Heidelberg. [Page 46]

Pearce, M. and Rohrmeier, M. (2012). Music Cognition and the Cognitive Sciences. *Topics in Cognitive Science*, 4(4):468–484. [Page 3]

Pearce, M. and Rohrmeier, M. (2018). Musical Syntax II: Empirical Perspectives. In Bader, R., editor, *Springer Handbook of Systematic Musicology*, pages 487–505. Springer Berlin Heidelberg, Berlin, Heidelberg. [Pages 39 and 46]

Pearce, M. T. (2005a). *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition.* PhD thesis, City University, London. [Pages 13 and 46]

Pearce, M. T. (2005b). *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition.* PhD Thesis, City University London. [Page 46]

Pearce, M. T. (2018). Statistical learning and probabilistic prediction in music cognition: Mechanisms of stylistic enculturation: Enculturation: Statistical learning and prediction. *Annals of the New York Academy of Sciences*, 1423(1):378–395. [Pages 13 and 46]

Pearce, M. T. and Wiggins, G. A. (2006). Expectation in Melody: The Influence of Context and Learning. *Music Perception*, 23(5):377–405. [Page 46]

Peel, J. and Slawson, W. (1984). Review of A generative theory of tonal music. *Journal of Music Theory*, 28:271–294. [Page 42]

Peeters, G. (2006). Musical key estimation of audio signal based on hidden Markov modeling of chroma vectors. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 127–131. Citeseer. [Page 46]

Pereira, F. C. and Warren, D. H. (1980). Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. *Artificial intelligence*, 13(3):231–278. [Page 76]

Perfors, A., Regier, T., and Tenenbaum, J. B. (2006). Poverty of the Stimulus? A Rational Approach. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 28. [Page 27]

Perfors, A., Tenenbaum, J. B., and Regier, T. (2011). The learnability of abstract syntactic principles. *Cognition*, 118(3):306–338. [Pages 2, 26, and 27]

Pfleiderer, M., Frieler, K., Abeßer, J., Zaddach, W.-G., and Burkhart, B. (2017). *Inside the Jazzomat: New Perspectives for Jazz Research.* Schott Music GmbH & Co. KG, Mainz, Germany. [Page 46]

Piattelli-Palmarini, M., editor (1980). *Language and Learning: The Debate between Jean Piaget and Noam Chomsky.* Harvard University Press, Cambridge, MA. [Page 26]

Piston, W. (1948). *Harmony.* W. W. Norton & Company, New York. [Page 46]

Pitsch, G. (1994). LL(k)-parsing of coupled-context-free grammars. *Computational intelligence*, 10(4):563–578. [Page 75]

Pitt, D. (2020). Mental representation. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, spring 2020 edition. [Page 23]

Pullum, G. K. and Scholz, B. C. (2002). Empirical assessment of stimulus poverty arguments. *The linguistic review*, 19(1-2):9–50. [Pages 2 and 27]

Quick, D. (2010). Generating music using concepts from Schenkerian analysis and chord spaces. *Yale University, Department of Computer Science, Tech. Rep. YALEU/DCS/RR-1440.* [Page 40]

Quick, D. (2014). *Kulitta: A Framework for Automated Music Composition.* Yale University. [Page 40]

Quick, D. (2016). Learning production probabilities for musical grammars. *Journal of New Music Research*, 45(4):295–313. [Pages 2 and 40]

Quick, D. and Hudak, P. (2013a). Grammar-based automated music composition in Haskell. In *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, pages 59–70. [Page 40]

Quick, D. and Hudak, P. (2013b). A temporal generative graph grammar for harmonic and metrical structure. In *ICMC.* [Page 40]

Quinn, I. and White, C. W. (2017). Corpus-derived key profiles are not transpositionally equivalent. *Music Perception: An Interdisciplinary Journal*, 34(5):531–540. [Page 118]

Rebuschat, P., Rohrmeier, M., Hawkins, J. A., and Cross, I., editors (2011). *Language and Music as Cognitive Systems.* Oxford University Press, New York. [Page 40]

Rizo, D. and Marsden, A. (2016). A standard format proposal for hierarchical analyses and representations. In *Proceedings of the 3rd International Workshop on Digital Libraries for Musicology*, pages 25–32. [Page 52]

Roads, C. and Wieneke, P. (1979). Grammars as representations for music. *Computer Music Journal*, pages 48–55. [Page 39]

Rodriguez Zivic, P. H., Shifres, F., and Cecchi, G. A. (2013). Perceptual basis of evolving Western musical styles. *Proceedings of the National Academy of Sciences*, 110(24):10034–10038. [Page 46]

Rohrmeier, M. (2011). Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53. [Pages 2, 7, 11, 16, 39, 52, and 84]

Rohrmeier, M. (2013). Musical Expectancy: Bridging Music Theory, Cognitive and Computational Approaches. *Zeitschrift der Gesellschaft für Musiktheorie [Journal of the German-speaking Society of Music Theory]*, 10(2):343–371. [Pages 7 and 13]

Rohrmeier, M. (2020a). The syntax of jazz harmony: Diatonic tonality, phrase structure, and form. *Music Theory and Analysis*, 7(1):1–63. [Pages 1, 2, 7, 10, 11, 13, 16, 17, 20, 22, and 39]

Rohrmeier, M. (2020b). Towards a Formalization of Musical Rhythm. In *21st International Society for Music Information Retrieval Conference*, Montréal, Canada. [Page 179]

Rohrmeier, M. and Cross, I. (2008). Statistical properties of tonal harmony in Bach's chorales. In *Proceedings of the 10th International Conference on Music Perception and Cognition*, volume 619, page 627. Hokkaido University Sapporo, Japan. [Page 46]

Rohrmeier, M. and Cross, I. (2009). Tacit tonality-Implicit learning of context-free harmonic structure. In *ESCOM 2009: 7th Triennial Conference of European Society for the Cognitive Sciences of Music*. [Page 24]

Rohrmeier, M. and Cross, I. (2013). Artificial grammar learning of melody is constrained by melodic inconsistency: Narmour's principles affect melodic learning. *PloS one*, 8(7):e66174. [Page 24]

Rohrmeier, M., Dienes, Z., Guo, X., and Fu, Q. (2014). Implicit learning and recursion. In *Language and Recursion*, pages 67–85. Springer, New York. [Page 24]

Rohrmeier, M., Fu, Q., and Dienes, Z. (2012). Implicit Learning of Recursive Context-Free Grammars. *PLoS ONE*, 7(10). [Page 24]

Rohrmeier, M. and Graepel, T. (2012). Comparing Feature-Based Models of Harmony. In *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval*, London. [Page 46]

Rohrmeier, M. and Neuwirth, M. (2015). Towards a Syntax of the Classical Cadence. In Neuwirth, M. and Bergé, P., editors, *What Is a Cadence? Theoretical and Analytical Perspectives on Cadences in the Classical Repertoire*. Leuven University Press. [Pages 2, 7, 11, 39, and 52]

Rohrmeier, M. and Pearce, M. (2018). Musical Syntax I: Theoretical Perspectives. In Bader, R., editor, *Springer Handbook of Systematic Musicology*, pages 473–486. Springer Berlin Heidelberg, Berlin, Heidelberg. [Pages 1, 39, and 46]

Rohrmeier, M. and Rebuschat, P. (2012). Implicit learning and acquisition of music. *Topics in cognitive science*, 4(4):525–553. [Pages 1, 23, 24, and 25]

Rohrmeier, M., Rebuschat, P., and Cross, I. (2011). Incidental and online learning of melodic structure. *Consciousness and Cognition*, 20(2):214–222. [Page 24]

Rohrmeier, M. and Widdess, R. (2017). Incidental Learning of Melodic Structure of North Indian Music. *Cognitive Science*, 41(5):1299–1327. [Page 24]

Rohrmeier, M., Zuidema, W., Wiggins, G. A., and Scharff, C. (2015). Principles of structure building in music, language and animal song. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370. [Page 7]

Rohrmeier, M. A. and Cross, I. (2014). Modelling unsupervised online-learning of artificial grammars: Linking implicit and statistical learning. *Consciousness and cognition*, 27:155–167. [Page 24]

Rohrmeier, M. A. and Koelsch, S. (2012). Predictive information processing in music cognition. A critical review. *International Journal of Psychophysiology*, 83(2):164–175. [Page 23]

Rothstein, W. (1989). *Phrase Rhythm in Tonal Music.* Schirmer Books, New York. [Page 21]

Rubin, D. B. (1981). The Bayesian Bootstrap. *The Annals of Statistics*, 9(1):130–134. [Page 126]

Saffran, J. R., Johnson, E. K., Aslin, R. N., and Newport, E. L. (1999). Statistical learning of tone sequences by human infants and adults. *Cognition*, 70(1):27–52. [Pages 1 and 24]

Salley, K. and Shanahan, D. T. (2016). Phrase Rhythm in Standard Jazz Repertoire: A Taxonomy and Corpus Study. *Journal of Jazz Studies*, 11(1):1–39. [Pages 21 and 63]

Salzer, F. (1952). *Structural Hearing: Tonal Coherence in Music*, volume 1. Dover Publications (1962), New York. [Pages 7 and 40]

Salzer, F. and Schachter, C. (1989). *Counterpoint in Composition: The Study of Voice Leading.* Columbia University Press. [Page 40]

Sarkar, A. (2001). Applying co-training methods to statistical parsing. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–8. Association for Computational Linguistics. [Page 52]

Sato, T. and Kameya, Y. (1997). PRISM: A language for symbolic-statistical modeling. In *IJCAI*, volume 97, pages 1330–1339. [Page 76]

Schachter, C. (1980). Rhythm and linear analysis: Durational reduction. *The Music Forum*, 5:197–232. [Page 21]

Schellenberg, E. G. (1996). Expectancy in melody: Tests of the implication-realization model. *Cognition*, 58(1):75–125. [Page 43]

Schenker, H. (1935). *Der Freie Satz. Neue Musikalische Theories and Phantasien.* Universal edition edition. [Pages 2, 7, and 40]

Schervish, M. J. (2012). *Theory of Statistics*. Springer Science & Business Media. [Page 32]

Schmuckler, M. A. (1989). Expectation in music: Investigation of melodic and harmonic processes. *Music Perception*, 7(2):109–149. [Page 13]

Schön, D., Boyer, M., Moreno, S., Besson, M., Peretz, I., and Kolinsky, R. (2008). Songs as an aid for language acquisition. *Cognition*, 106(2):975–983. [Page 24]

Sears, D. R., Arzt, A., Frostel, H., Sonnleitner, R., and Widmer, G. (2017). Modeling harmony with skip-grams. In *18th International Society for Music Information Retrieval Conference*, Suzhou, China. [Page 46]

Sears, D. R., Pearce, M. T., Caplin, W. E., and McAdams, S. (2018). Simulating melodic and harmonic expectations for tonal cadences using probabilistic models. *Journal of New Music Research*, 47(1):29–52. [Page 13]

Sears, D. R. and Widmer, G. (2020). Beneath (or beyond) the surface: Discovering voice-leading patterns with skip-grams. *Journal of Mathematics and Music*. [Page 46]

Seifert, U. (1993). *Systematische Musiktheorie Und Kognitionswissenschaft: Zur Grundlegung Der Kognitiven Musikwissenchaft*. Number 69. Verlag Für Systematische Musikwissenschaft, Bonn, Germany. [Page 3]

Seki, H., Matsumura, T., Fujii, M., and Kasami, T. (1991). On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191—229. [Page 76]

Shaffer, K., Vasiete, E., Jacquez, B., Davis, A., Escalante, D., Hicks, C., McCann, J., Noufi, C., and Salminen, P. (2020). A cluster analysis of harmony in the McGill Billboard dataset. *Empirical Musicology Review*, 14(3-4):146. [Page 46]

Shanahan, D. and Broze, Y. (2012). A Diachronic Analysis of Harmonic Schemata in Jazz. In *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, Thessaloniki, Greece. [Pages 46 and 59]

Sheh, A. and Ellis, D. P. W. (2003). Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the 4th International Conference on Music Information Retrieval*, Baltimore, MD. Johns Hopkins University. [Page 46]

Sher, C., editor (1991). *The New Real Book Volume Two*. Sher Music Co., Petaluma, CA. [Page 10]

Sher, C., editor (1995). *The New Real Book Volume Three*. Sher Music Co., Petaluma, CA. [Page 113]

Shieber, S. M., Schabes, Y., and Pereira, F. C. (1995). Principles and implementation of deductive parsing. *The Journal of logic programming*, 24(1-2):3–36. [Page 79]

Sidorov, K. A., Jones, A., and Marshall, A. D. (2014). Music Analysis as a Smallest Grammar Problem. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 301—306. [Page 40]

Sigtia, S., Boulanger-Lewandowski, N., and Dixon, S. (2015). Audio Chord Recognition with a Hybrid Recurrent Neural Network. In *ISMIR*, pages 127–133. [Page 47]

Sikora, F. (2003). *Neue Jazz-Harmonielehre. Verstehen, Hören, Spielen: Von Der Theorie Zur Improvisation.* Mainz, Germany. [Page 9]

Smither, S. R. (2020). *Conceptualizing Tunes: Avant-Textes, Referents, and the Analysis of Musical Structure in Jazz.* PhD thesis, The State University of New Jersey. [Page 9]

Smoliar, S. W. (1979). A computer aid for Schenkerian analysis. In *Proceedings of the 1979 Annual Conference*, pages 110–115. [Page 41]

Stabler, E. (1996). Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer. [Page 76]

Stabler, E. P. (2011). *Computational Perspectives on Minimalism.* Oxford University Press. [Page 76]

Steedman, M. (1996). The Blues and the Abstract Truth: Music and Mental Models. In *Mental Models in Cognitive Science: Essays in Honour of Phil Johnson-Laird*, pages 305–318. Psychology Press, Hove, UK. [Pages 1, 7, and 44]

Steedman, M. (2000). *The Syntactic Process.* MIT Press, Cambridge, MA. [Pages 44 and 181]

Steedman, M. (2019). Combinatory Categorial Grammar. In Kertész, A., Moravcsik, E., and Rákosi, C., editors, *Current Approaches to Syntax*, pages 389–420. De Gruyter, Berlin, Boston. [Page 44]

Steedman, M. and Baldridge, J. (2006). Combinatory Categorial Grammar. In Brown, K., editor, *Encyclopedia ofLanguage and Linguistics*, volume 2, pages 610–622. Elsevier, Oxford, second edition. [Page 44]

Steedman, M. J. (1977). The Perception of Musical Rhythm and Metre. *Perception*, 6(5):555–569. [Page 44]

Steedman, M. J. (1984). A generative grammar for jazz chord sequences. *Music Perception*, 2(1):52–77. [Page 44]

Stern, M. A. (1858). über eine zahlentheoretische Funktion. *Journal für die reine und angewandte Mathematik*, 55:193–220. [Page 164]

Stock, J. (1993). The application of Schenkerian Analysis to Ethnomusicology: Problems and possibilities. *Music Analysis*, 12(2):215–240. [Page 41]

Stolcke, A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational linguistics*, 21(2):165–201. [Pages 79 and 88]

Strunk, S. (1979). *The Harmony of Early Bop: A Layered Approach*, volume 6. Journal of Jazz Studies. [Pages 14 and 21]

Sundberg, J. and Lindblom, B. (1976). Generative theories in language and music descriptions. *Cognition*, 4:99–122. [Pages 2 and 40]

Sundberg, J. and Lindblom, B. (1991). Generative Theories for Describing Musical Structure. In Howell, P., West, R., and Cross, I., editors, *Representing Musical Structure*, pages 242–272. Academic Press, London. [Page 39]

Temperley, D. (2000). The Line of Fifths. *Music Analysis*, 19(3):289–319. [Page 112]

Temperley, D. (2007). *Music and Probability.* MIT Press, Cambridge, MA. [Page 29]

Temperley, D. (2011). Composition, perception, and Schenkerian theory. *Music Theory Spectrum*, 33(2):146—168. [Pages 7 and 41]

Temperley, D. (2012). Computational models of music cognition. In Deutsch, D., editor, *The Psychology of Music*, pages 327—368. Elsevier, Amsterdam. [Pages 25 and 26]

Temperley, D. (2018). *The Musical Language of Rock.* Oxford University Press, Oxford. [Page 46]

Temperley, D. and de Clercq, T. (2013). Statistical analysis of harmony and melody in rock music. *Journal of New Music Research*, 42(3):187–204. [Page 46]

Tenenbaum, J. B., Griffiths, T. L., and Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. page 10. [Page 33]

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., and Goodman, N. D. (2011). How to Grow a Mind: Statistics, Structure, and Abstraction. *Science*, 331(6022):1279–1285. [Pages 2 and 24]

Thagard, P. (2019). Cognitive science. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2019 edition. [Page 25]

Tidhar, D. (2005). *A Hierarchical and Deterministic Approach to Music Grammars and Its Application to Unmeasured Preludes.* dissertation.de, Berlin. [Page 40]

Tillmann, B. (2005). Implicit investigations of tonal knowledge in nonmusician listeners. *Annals of the New York Academy of Sciences*, 1060(1):100–110. [Page 24]

Tillmann, B. and McAdams, S. (2004). Implicit learning of musical timbre sequences: Statistical regularities confronted with acoustical (dis) similarities. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(5):1131. [Page 24]

Tojo, S., Oka, Y., and Nishida, M. (2006). Analysis of chord progression by HPSG. In Devedzic, V., editor, *IASTED International Conference on Artificial Intelligence and Applications, Part of the 24th Multi-Conference on Applied Informatics, Innsbruck, Austria, February 13-16, 2006*, pages 305–310. IASTED/ACTA Press. [Page 40]

Tsushima, H., Nakamura, E., Itoyama, K., and Yoshii, K. (2017). Function- and Rhythm-Aware Melody Harmonization Based on Tree-Structured Parsing and Split-Merge Sampling of Chord Sequences. page 7. [Page 149]

Tsushima, H., Nakamura, E., Itoyama, K., and Yoshii, K. (2018). Generative statistical models with self-emergent grammar of chord sequences. *Journal of New Music Research*, 47(3):226–248. [Pages 2 and 149]

Tsushima, H., Nakamura, E., and Yoshii, K. (2020). Bayesian Melody Harmonization Based on a Tree-Structured Generative Model of Chord Sequences and Melodies. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28. [Pages 2 and 40]

Tymoczko, D. (2006). The geometry of musical chords. *Science*, 313(5783):72–74. [Pages 9 and 46]

Tymoczko, D. (2011). *A Geometry of Music: Harmony and Couterpoint in the Extended Common Practice.* Oxford University Press, Oxford. [Pages 9, 22, and 46]

Vijay-Shanker, K. and Weir, D. (1993). Parsing some constrained grammar formalisms. *Computational Linguistics*, 19(4):591–636. [Page 77]

Vijay-Shanker, K., Weir, D., and Joshi, A. K. (1987). Characterizing structural descriptions produced by various grammatical formalisms. In *25th Meeting of the Association for Computational Linguistics.* [Page 76]

Vijay-Shanker, K. and Weir, D. J. (1994). The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory*, 27(6):511–546. [Page 76]

Wagenmakers, E.-J., Marsman, M., Jamil, T., Ly, A., Verhagen, J., Love, J., Selker, R., Gronau, Q. F., Šmíra, M., and Epskamp, S. (2018). Bayesian inference for psychology. Part I: Theoretical advantages and practical ramifications. *Psychonomic bulletin & review*, 25(1):35–57. [Page 125]

Wall, L., Lieck, R., Neuwirth, M., and Rohrmeier, M. (2020). The Impact of Voice Leading and Harmony on Musical Expectancy. *Scientific Reports*, 10(1):1–8. [Page 22]

Weir, D. J. (1988). *Characterizing Mildly Context-Sensitive Grammar Formalisms.* PhD thesis, University of Pennsylvania. [Page 76]

Wetherell, C. S. (1980). Probabilistic Languages: A Review and Some Open Questions. *ACM Computing Surveys (CSUR)*, 12(4):361–379. [Page 74]

Whorley, R. P., Wiggins, G. A., Rhodes, C., and Pearce, M. T. (2013). Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonization problem. *Journal of New Music Research*, 42(3):237–266. [Page 46]

Wiggins, G. A. (2011). Computer models of (music) cognition. In Rebuschat, P., Rohrmeier, M., Hawkins, J. A., and Cross, I., editors, *Language and Music as Cognitive Systems*. Oxford University Press, New York. [Page 25]

Wiggins, G. A., Müllensiefen, D., and Pearce, M. T. (2010). On the non-existence of music: Why music theory is a figment of the imagination. *Musicae Scientiae*, pages 231–255. [Page 1]

Wiggins, G. A., Pearce, M. T., and Müllensiefen, D. (2011). Computational modeling of music cognition and musical creativity. In Dean, R. T., editor, *The Oxford Handbook of Computer Music*. Oxford University Press. [Page 25]

Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *journal of Music Theory*, 12(1):2–49. [Pages 2 and 40]

Wu, S.-L. and Yang, Y.-H. (2020). The Jazz Transformer on the Front Line: Exploring the Shortcomings of AI-composed Music through Quantitative Measures. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*. [Page 47]

Yamamoto, H., Uehara, Y., and Tojo, S. (2020). Jazz Harmony Analysis with epsilon-Transition and Cadential Shortcut. In *Proceedings of the 17th Sound and Music Computing Conference*, Torino. [Page 43]

Yoshinaka, R. (2011). Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theoretical Computer Science*, 412(19):1821–1831. [Page 149]

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n3. *Information and control*, 10(2):189–208. [Page 79]

Yust, J. (2015). Voice-Leading Transformation and Generative Theories of Tonal Structure. *Music Theory Online*, 21(4). [Page 41]

Yust, J. (2018). *Organized Time: Rhythm, Tonality, and Form*. Oxford University Press. [Page 42]

Yust, J. (2019). Stylistic information in pitch-class distributions. *Journal of New Music Research*, 48(3):217–231. [Page 46]

Yust, J. D. (2006). *Formal Models of Prolongation*. PhD thesis, University of Washington, Washington. [Page 41]

Zhou, X. and Lerch, A. (2015). Chord detection using deep learning. In *Proceedings of the 16th ISMIR Conference*, volume 53, page 152. [Page 47]

Daniel Harasim

Contact Information

EPFL CDH DHI DCML
INN 115, Station 14
1015 Lausanne
Switzerland

daniel.harasim@epfl.ch

Research Interests

Mathematical and computational modeling of musical structures. Computational cognitive science. Music cognition. Computational Musicology. Bayesian learning models. Probabilistic machine learning. Probabilistic formal language theory. Composition and improvisation in Jazz. Mathematical music theory.

Employment

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland — since 2020
Postdoctoral Researcher: Digital and Cognitive Musicology Lab (DCML)

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland — 2017–2020
Doctoral Assistant: Digital and Cognitive Musicology Lab (DCML)

Technische Universität Dresden (TUD), Germany — 2015–2017
Doctoral Assistant: Dresden Music Cognition Lab (DMCL)

Education

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland — 2017–2020
PhD Student: Digital and Cognitive Musicology Lab (DCML)

McGill University, Montréal, Canada — Jan–Apr 2017
Visiting Scholar: Department of Linguistics

Technische Universität Dresden (TUD), Germany — 2015–2017
PhD Student: Dresden Music Cognition Lab (DMCL)

Technische Universität Dresden (TUD), Germany — 2012–2015
Master of Science, Mathematics

Technische Universität Dresden (TUD), Germany — 2009–2012
Bachelor of Science, Mathematics

Dr.-Wilhelm-André-Gymnasium, Chemnitz, Germany — 2008
Abitur (High-School Diploma)

Grants and Awards

great!pid4all traval grant (group2group exchange for academic talents) — 2017
Deutscher Akademischer Austauschdienst (DAAD)

Theses

The Learnability of the Grammar of Jazz:
Bayesian Inference of Hierarchical Structures in Harmony — 2020
PhD Thesis
Supervisor: Martin Rohrmeier (EPFL)
Co-supervisor: Timothy J. O'Donnell (McGill University)

Voice Leading and Circular Spaces of Stacked Harmonies in the Framework of Tone Structures 2015
Master Thesis
Supervisor: Stefan E. Schmidt (TUD)

Grundlegende Elemente einer extensionalen Standardsprache der Jazz-Harmonielehre 2012
Bachelor Thesis
Supervisor: Stefan E. Schmidt (TUD)

Refereed Conference and Journal Articles

Harasim D, Moss FC, Ramirez M, and Rohrmeier M (accepted) Exploring the foundations of tonality: Statistical cognitive modeling of modes in the history of Western classical music. *Humanities and Social Sciences Communications.*

Harasim D, Finkensiep C, Ericson P, O'Donnell TJ, and Rohrmeier M (2020) The Jazz Harmony Treebank. *Proceedings of the 21th International Society for Music Information Retrieval Conference.* `http://doi.org/10.5281/zenodo.4245406`

Harasim D, Schmidt SE, and Rohrmeier M (2020) Axiomatic scale theory. *Journal of Mathematics and Music.*
`https://doi.org/10.1080/17459737.2019.1696899`

Harasim D, O'Donnell TJ, and Rohrmeier M (2019) Harmonic Syntax in Time: Rhythm Improves Grammatical Models of Harmony. *Proceedings of the 20th International Society for Music Information Retrieval Conference.*
`http://archives.ismir.net/ismir2019/paper/000039.pdf`

Moss, FC, Neuwirth M, **Harasim D**, and Rohrmeier M (2019) Statistical characteristics of tonal harmony: A corpus study of Beethoven's string quartets. *PLOS ONE*, 14(6).
`https://doi.org/10.1371/journal.pone.0217242`

Harasim D, Noll T, and Rohrmeier M (2019) Distant Neighbors and Interscalar Contiguities. In M. Montiel, F. Gomez-Martin, and O. A. Agustín-Aquino (Eds.), *Mathematics and Computation in Music* (Vol. 11502, pp. 172–184). Springer International Publishing.
`https://doi.org/10.1007/978-3-030-21392-3_14`

Harasim D, Rohrmeier M, and O'Donnell TJ (2018) A Generalized Parsing Framework for Generative Models of Harmonic Syntax. *Proceedings of the 19th International Society for Music Information Retrieval Conference.*
`http://ismir2018.ircam.fr/doc/pdfs/258_Paper.pdf`

Neuwirth M, **Harasim D**, Moss FC, and Rohrmeier M (2018) The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets. *Frontiers in Digital Humanities*, 5.
`https://doi.org/10.3389/fdigh.2018.00016`

Harasim D, Schmidt SE, and Rohrmeier M (2016) Bridging scale theory and geometrical approaches to harmony: The voice-leading duality between complementary chords. *Journal of Mathematics and Music*, 10(3), 193–209.
`https://doi.org/10.1080/17459737.2016.1216186`

Talks and Presentations

Harmonic Syntax in Time 06.11.2019
20th International Society for Music Information Retrieval Conference
Delft, The Netherlands

Harmonic syntax: A Uniform Framework for Elaborations, Substitutions, and Form 30.08.2019
Computational Approaches in Language and Music Cognition
Research Workshop, University of Cologne
Cologne, Germany

Harmonic Syntax in Time 11.07.2019
Department of Linguistics, McGill University
Montréal, Canada

Semiring Parsing and Variational Inference for Abstract Context-Free Grammars 08.07.2019
Department of Linguistics, McGill University
Montréal, Canada

A Data-Driven History of Major and Minor 02.07.2019
Corpus Research as a Means of Unlocking Musical Grammar
International Research Workshop
Tel-Aviv, Israel

Distant Neighbors and Interscalar Contiguities 19.06.2019
Mathematics and Computation in Music Conference
Madrid, Spain

Unsupervised Grammar Induction of Jazz Harmony 25.03.2019
SEMPRE Graduate Conference 2019
Cambridge, UK

Unsupervised Grammar Induction of Jazz Harmony 22.02.2019
First Swiss Digital Humanities Student Exchange (DHX2019)
Basel, Switzerland

A Generalized Parsing Framework for Generative Models of Harmonic Syntax 24.09.2018
19th International Society for Music Information Retrieval Conference
Paris, France

Computational Inference of Syntactic Structures in Music 11.06.2018
European Research Music Conference
Barcelona, Spain

A Brief History of Tonality 23.01.2018
Applied Machine Learning Days
Lausanne, Switzerland

Analyzing the Syntax of Double-Plagal and Blues Progressions in Rock, Pop, and Jazz 18.11.2017
17. Jahreskongress der Gesellschaft für Musiktheorie (GMTH) and
27. Arbeitstagung der Gesellschaft für Popularmusikforschung (GfPM)
Graz, Austria

Beethoven's String Quartets: Introducing an XML-Based Corpus of Harmonic Labels Using a New Annotation System 17.05.2017
Music Encoding Conference
Tours, France

Musical Syntax 28.03.2017
Music Perception and Cognition Laboratory
Schulich School of Music, McGill University
Montréal, Canada

Generalized Context-Free Parsing Workshop 14.03.2017
Department of Linguistics, McGill University
Montréal, Canada

Musical Syntax 15.02.2017
Reasoning and Learning Lab
School of Computer Science, McGill University
Montréal, Canada

Musical Syntax 09.02.2017
Department of Linguistics, McGill University
Montréal, Canada

Musical Scales as Embeddings of Circular Ordered Sets 27.05.2016
Faculty of Mathematics, University of Regensburg
Regensburg, Germany

A Generalized Probabilistic Parser for Musical Purposes 23.01.2016
Symposium Towards a World Music Theory, University of Hamburg
Hamburg, Germany

The Geometry of Minimal Voice Leading 15.06.2015
Institute of Art and Music, Technische Universität Dresden
Dresden, Germany

Teaching and Supervision

Visual Hierarchical Analysis of Tonality using the Discrete Fourier Transform 2019–2020
Master Thesis
Co-supervisor

Digital Musicology 2018–2019
Graduate Course incl. Student Projects
Teaching Assistant

Music, Musical Structure, Artificial Neural Networks, and the Mind 2019
Sommerakademie der Schweizer Studienstiftung (Summer School)
Teacher

Musical improvisation, invention and creativity 2019
Undergraduate Course
Teaching Assistant

Implementation of an Online Platform for Behavioral Experiments in Music Psychology 2018
Student Project
Co-supervisor

Mathematics, Music, and Cognition 2015–2017
Graduate Seminar
Seminar Facilitator

Programming for Musicologists 2016
Graduate Course
Teaching Assistant

Interdisciplinary Student Projects in Technical Design and Musicology 2015–2016
Co-supervisor

Linear algebra for mathematicians 2011–2014
Undergraduate course
Teaching Assistant

Mathematical methods for computer scientists 2013
Undergraduate course
Teaching Assistant

Linear algebra for computer scientists 2012–2013
Undergraduate course
Teaching Assistant

Skills

Natural Languages
German (native), English (proficient), French (basic)

Programming Languages
Haskell, Python, Julia, ClojureScript, Rust, Latex

Music Instruments
Upright Bass (main instrument), Guitar (first instrument), Piano (basic skills)