On Bisimulations for the Spi Calculus

EPFL I&C Technical Report IC/2003/34[*]

Johannes Borgström
EPFL, Switzerland

Uwe Nestmann[†]
EPFL, Switzerland

May 23, 2003, 19:35

Abstract

The spi calculus is an extension of the pi calculus with cryptographic primitives, designed for the verification of cryptographic protocols. Due to the extension, the naive adaptation of labeled bisimulations for the pi calculus is too strong to be useful for the purpose of verification. Instead, as a viable alternative, several "environment-sensitive" bisimulations have been proposed. In this paper we formally study the differences between these bisimulations.

1 Introduction

The spi calculus, proposed by [AG99] as an extension of the pi calculus [Mil99], is a process calculus designed for the description and formal verification of cryptographic protocols.

According to [AG99], many correctness properties for cryptographic protocols are naturally expressed through may-testing equivalences between certain process terms, but proofs of such properties are notoriously hard due to the requirement of infinitary quantifications (usually quantifications over infinitely many process contexts). In contrast, the standard bisimulation-based notion of equivalence [Par81] on process terms provides a coinductive proof technique, usually avoiding infinitary quantifications. The interest in bisimulation notions for the spi calculus follows from the fact that bisimilarity (usually defined as the largest bisimulation) is a sound, but not complete, approximation to may-testing equivalence.

Bisimilarity is an observational equivalence, based on the idea of an environment observing a pair of processes to see whether it may distinguish one from the other. The environment typically observes the labeled transition system derived from the operational semantics of processes. In usual process calculi, the two points of view of an observing environment and of an observed process are symmetric: any transition that a process can do according to its semantics is also observable by the environment. This symmetry is no longer valid in the case of the spi calculus.

In Figure 1, we highlight some meaningful and meaningless transitions in the spi calculus compared to the pi calculus. The various labels represent the input $a\,b$ of a name b along channel a, the output $\overline{a}\,b$ of a message b along channel a, or the bound counterpart $(\nu b)\,\overline{a}\,b$ for a fresh name b, and the (bound) output $(\nu bk)\,\overline{a}\langle \mathrm{E}_k(b)\rangle$ along channel a of a message $\mathrm{E}_k(b)$ representing the encryption of cleartext b using key k. The displayed transitions represent the possible observations about a process from the point of view of an environment interacting with the process. For example, an environment observing P might see the bound output $(\nu b)\,\overline{a}\,b$, which at the same time means that the environment itself performs a respective input operation. After the transition $P \xrightarrow{(\nu b)\,\overline{a}\,b} P'$, the fresh name b received by the environment may be used to interact with P' as seen in Figure 1. Essentially, once the environment receives a name, it may freely use it in interactions. The situation is different in the spi calculus, where exchanged messages may be

[*] Supported by the Swiss National Science Foundation, grant No. 21-65180.01.

[†] Uwe.Nestmann@EPFL.ch

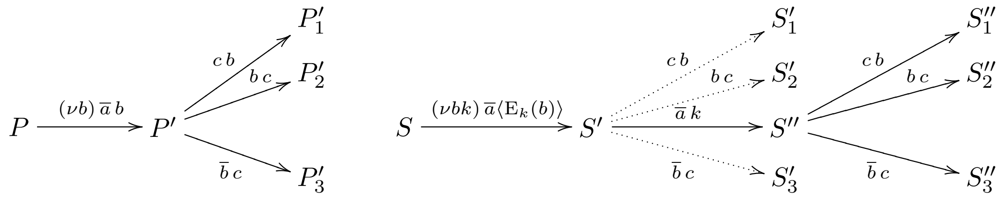

Figure 1: Environment Transitions in pi and spi calculi

encrypted, as in the transition from S to S'. Note that both the key k and the datum b are bound, but when S transmits the encrypted message, neither the key nor the clear text b are accessible to the environment. Therefore, none of the dotted transitions are possible for S' in interaction with an environment: since the environment does not know the key k itself, it cannot interact with S' using the cleartext (the name b) hidden inside the ciphertext $\mathrm{E}_k(b)$, neither to communicate *on* the channel b nor to send back b to the process. However, this becomes possible when the key k is sent in the clear to the environment, as in the transition from S' to S''.

Summing up the previous examples, a proper treatment of transitions with respect to bisimulation in the spi calculus must, in contrast to the pi calculus, explicitly take into account the *knowledge* of an environment about a process. As a means to capture this environment knowledge, the notion of *environment-sensitive* bisimulation has been developed for the spi calculus, in various styles:

- [AG98] introduced *framed* bisimulation by imposing on every bisimulation pair a common *frame-theory* pair that represents the knowledge of the environment about the pair. The frame is the set of names (channels, keys) that the environment has learned so far, while the theory is the set of pairs of non-name data items received from the pair of processes during the bisimulation game that the environment must believe to be "the same", because it has no means (i.e., decryption keys) to distinguish them.
- [BDP99] introduced another notion under the generic name of environment-sensitive bisimulation. Here, each of the processes in a bisimulation pair is accompanied by an environment, which (roughly) lists the messages received from the process in the past. In this paper, we call their variant *alley* bisimulation, pictorially reminding of the separate environments. To express the identification of non-distinguishable data items, an explicit condition of *equivalence* on the environments is imposed.
- [EHHO99] introduced *fenced* bisimulation, an approximation to framed bisimulation by getting rid of one of its infinitary quantifications.

All of the above notions of bisimulation are, assuming that the observing environments know all free names of the related processes, sound approximations of may-testing equivalence.

Comparing Bisimulations

The immediate question on the various competing notions of bisimulation is (1) how they relate to each other, and (2) how each of them relates to *barbed equivalence*, which is a uniformly defined contextual notion of bisimulation that is usually considered prime among all bisimilarities [MS92]. So far, these questions have only been treated in parts. [BDP99] proved that alley bisimilarity, for a wide range of processes, is a sound and complete approximation of barbed equivalence. In contrast, [AG98] were already aware that their notion of framed bisimulation is strictly stronger than barbed equivalence, at least when the spi calculus contains a pairing construct; without pairing, no result has been published. [EHHO99] proposed that fenced bisimulation would coincide with framed bisimulation, but their proof is flawed (see below): framed bisimilarity is not contained in fenced bisimilarity. Finally, [FHJ01] recently showed that fenced bisimulation coincides with a strengthened form of alley bisimulation (also treated by [BDP99]; here, we call it "trellis" bisimulation).

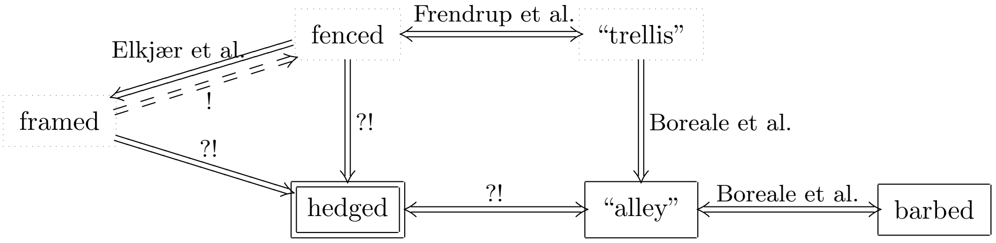

Figure 2: Comparing Bisimulations

All the above bisimulations exist in both early and late variants. As only the early alley bisimulation is complete with respect to barbed equivalence we study the early variants of all bisimulations. However, the results of this paper should apply without modifications to late bisimulations.

In Figure 2, we pictorially summarize the various relations, which lets us also clearly discuss the scientific contribution of the current paper.

Contributions

In this paper, we formally highlight the differences between the above-mentioned environment-sensitive bisimulations by introducing an improved form of framed bisimulation, called *hedged* bisimulation. In summary, all of the question marks in Figure 2 are supported by proofs for positive results and counterexamples for negative results. In particular, we prove that hedged bisimulation coincides with alley bisimulation, and thus also with barbed equivalence. However, being defined in the style of framed and fenced bisimulation, hedged bisimulation allows us to formally and much more intuitively assess the differences of the former notions, as indicated in Figure 2. We also exhibit, by means of a counterexample, that fenced bisimulation is not complete w.r.t. framed bisimulation, contradicting the results of [EHHO99]. To further clarify the structural differences between the bisimilarities, we first describe the bisimilarities as categories. Then, we attempt to relate these categories in terms of embedding functors and equivalences.

Related Work

The spi calculus fragment studied in this paper has symmetric encryption and pairing as constructors for composite messages. A much more general setting, with arbitrary term equivalence and constructors was studied by [AF01]. There, the focus is on manual studies, as term equivalence may turn out to be undecidable due to the underlying equational theory. Environment knowledge is represented by substitutions, resembling the environments of alley bisimulation.

[Cor02] extends alley bisimulation to a more general calculus. The additions include public key encryption, composite keys, arbitrary regular guards and function symbols. In spite of this involved calculus, environment equivalence is decidable because of the regularity of the guards and an equational theory that only admits functions that are either one-way, completely invertible or partially invertible (i.e., encryption).

[Hüt02] has shown that framed bisimilarity is decidable for finite spi, a spi calculus with pairing but without replication nor recursion. In that paper, there is also an example showing that finite-control (i.e., recursion, parallel composition only at the top level) spi calculus is Turing-powerful, as opposed to finite-control pi calculus.

Overview

The spi calculus used in this paper is presented in Section 2. Definitions of the environment-sensitive bisimulations can be found in Section 3. In Section 4 we exhibit some examples showing the differences between framed, fenced and hedged bisimulation. A framework for relating environment-sensitive bisimilarities is laid in Section 5. In Section 6 we proceed by comparing the different ways of defining environments. The main work is done in Section 7, where we show that hedged bisimulation is equivalent to alley bisimulation and compare hedged bisimulation to framed and fenced bisimulation. In Section 8 we compare the bisimulations as categories, and express some up-to techniques in this setting.

2 Language

The spi calculus used in this paper is the one used by [BDP02], with some changes in notation. Furthermore, we build on the same assumptions on the underlying system of shared-key cryptography, which read as follows:

1. *Perfect Encryption*: A ciphertext $\mathrm{E}_k(M)$, i.e., a message M encrypted under a key k, can only be decrypted using k. The only way to produce the ciphertext $\mathrm{E}_k(M)$ is to encrypt M under k. If k is secret, no attacker can guess or forge k.

2. There is enough redundancy in the structure of messages to tell whether decryption of a message with a given key has actually succeeded or not.

3. There is enough redundancy in the structure of messages to tell their role (name or compound ciphertext).

4. The only way to form a new key is to get a fresh name from a primitive set of *names*.

Assumption 3 is necessary since we only permit communications on channels corresponding to a name (i.e., a encrypted message cannot be used as a channel). To explicitly check for this distinction we have a guard $\mathit{is\_name}(\delta)$, the semantics of which can be found in Table 2 on page 6.

2.1 Syntax

We assume a countably infinite set $\mathcal{N}$ of names. Names are untyped, meaning that the same name can be used as a channel, a key, a variable or the clear-text of a message. The lower case letters a, b, c, k, l, m, n, x, y, z are used to range over names. In examples, a, b, c are used for channels, k, l for keys, m, n for messages and x, y, z for variables when their subsequent usage is not explicited.

The syntax of expressions, guards, and processes, is given in Table 1.

In contrast to the pi calculus, the spi calculus offers next to mere names another kind of transmissible messages, namely *ciphertexts*, which are provided by the addition of primitive constructs to encrypt ($\mathrm{E}_\cdot(\cdot)$) and decrypt ($\mathrm{D}_\cdot(\cdot)$) data using a shared-key cryptographic system. Encryptions can be arbitrarily nested, but (in contrast to [AG99]) only proper names can be used as encryption keys. While expressions ζ are formed arbitrarily using the encryption and decryption operators, messages M represent proper decryption-free ciphertexts where the encryption keys are names. The role of decryption-free expressions δ is to formalize that decryption constructors can only occur within let-constructs, as indicated by the occurrences of ζ. This property will be preserved by the operational semantics later on.

Logical formulae ϕ generalize the usual equality operator of the pi calculus by conjunction and negation. Moreover, the predicate $\mathit{is\_name}(\delta)$ tests for the format of δ, i.e., whether it is a plain name or not. The formula let $z = \zeta$ in ϕ binds *the value of* the expression ζ, computed by evaluation as defined in Subsection 2.2, to the name z within formula ϕ.

Processes are formed as in the pi calculus, except for the following aspects:

- Input and output forms have to take into account that the channel and message positions might be (decryption-free) expressions; however, in a channel position only names make sense, otherwise the process will be stuck.
- Guarded processes generalize the standard matching construct.

$a, b, c \dots, k, l, m, n \dots, x, y, z$			names $\mathcal{N}$
ζ, η	$::=$	$a \quad \mid \quad \mathrm{E}_\zeta(\zeta) \quad \mid \quad \mathrm{D}_\zeta(\zeta)$	expressions $\mathcal{E}$
δ	$::=$	$a \quad \mid \quad \mathrm{E}_\delta(\delta)$	decryption-free expressions $\mathcal{D}$
M, N	$::=$	$a \quad \mid \quad \mathrm{E}_k(M)$	messages $\mathcal{M}$
ϕ, ψ	$::=$	$tt \quad \mid \quad \phi \wedge \phi \quad \mid \quad \neg\phi$	guards $\mathcal{G}$
	$\mid$	let $z = \zeta$ in ϕ	(decryption)
	$\mid$	$is\_name(\delta)$	(is a name)
	$\mid$	$[\,\delta = \delta\,]$	(equality)
P, Q	$::=$	$\mathbf{0}$	processes $\mathcal{P}$
	$\mid$	$\delta(x).P$	(input prefix)
	$\mid$	$\overline{\delta}\langle\delta\rangle.P$	(output prefix)
	$\mid$	$P + P$	(choice)
	$\mid$	$P \mid P$	(parallel)
	$\mid$	$(\nu a)\, P$	(restriction)
	$\mid$	$!\, P$	(replication)
	$\mid$	ϕP	(boolean guard)
	$\mid$	let $x = \zeta$ in P	(decryption)

Table 1: Syntax of the spi calculus

- There is a let-construct, both in guards and in processes, which is the only place where message decomposition occurs.

For the main text we leave out the treatment of tuples in messages; see the Appendix for details on this extension.

Free and bound names of terms are inductively defined as expected: a is bound in "$(\nu a)\, P$", x is bound in "$\delta(x).P$", in "let $x = \zeta$ in P", and in "let $x = \zeta$ in ϕ". Two processes are *α-equivalent* if they can be made equal by conflict-free renaming of bound names. Substitutions σ are mappings $\{^M/_x\}$ from names x to messages M. Substitutions are applied to processes, expressions and guards in the straightforward way, obeying the usual assumption that name-capture is avoided through implicit α-conversion: for example, $P\{^M/_x\}$ replaces all free occurrences of x in P by M, renaming bound names in P where needed.

2.2 Semantics

To define operational semantics we need to be able to evaluate both expressions and boolean guards. The evaluation function for expressions $[\![\,\cdot\,]\!] : \mathcal{E} \to \mathcal{M} \cup \{\bot\}$ and guards $[\![\,\cdot\,]\!] : \mathcal{G} \to \{tt, \mathit{ff}\}$ is defined recursively according to Table 2.

The operational semantics (see Table 3) follows [BDP02]. It uses an early input style semantics, which is standard from the pi calculus, except for the rules (GUARD) and (LET) which provide the only attempts to decrypt messages. A process ϕP behaves like P provided that ϕ evaluates to true; otherwise, ϕP is stuck. A process let $z = \zeta$ in P behaves like $P\{^{[\![\zeta]\!]}/_z\}$ provided that the evaluation of ζ succeeds; otherwise, let $z = \zeta$ in P is stuck. Symmetric variants of (PAR), (COM) and (SUM) are not explicited. By (ALP) we have that α-equivalent processes have the same transitions, so all relations on processes based on transitions will also be defined up to α-equivalence.

3 Environment-Sensitive Bisimulations

As motivated in the Introduction, bisimulations in the spi calculus must take into account the knowledge of the observing environment—potentially any kind of malicious attacker—at any moment in time. Since the interaction between an environment and a process is fully described by the exchange of messages, it is important to spell out the power of attackers in the spi calculus model.

$$
\begin{array}{lcl}
\llbracket\, a \,\rrbracket & = & a \\
\llbracket\, \mathrm{E}_{\zeta}(\eta) \,\rrbracket & = & \begin{cases} \mathrm{E}_{k}(M) & \text{if } \llbracket\, \eta \,\rrbracket = M \in \mathcal{M} \text{ and } \llbracket\, \zeta \,\rrbracket = k \in \mathcal{N} \\ \bot & \text{otherwise} \end{cases} \\
\llbracket\, \mathrm{D}_{\zeta}(\eta) \,\rrbracket & = & \begin{cases} M & \text{if } \llbracket\, \eta \,\rrbracket = \mathrm{E}_{k}(M) \in \mathcal{M} \text{ and } \llbracket\, \zeta \,\rrbracket = k \in \mathcal{N} \\ \bot & \text{otherwise} \end{cases} \\
\llbracket\, tt \,\rrbracket & = & tt \\
\llbracket\, \phi \wedge \psi \,\rrbracket & = & \llbracket\, \phi \,\rrbracket \wedge \llbracket\, \psi \,\rrbracket \\
\llbracket\, \neg\psi \,\rrbracket & = & \neg\llbracket\, \psi \,\rrbracket \\
\llbracket\, \text{let } z = \zeta \text{ in } \phi \,\rrbracket & = & \begin{cases} \llbracket\, \phi\{^{M}\!/_{z}\} \,\rrbracket & \text{if } \llbracket\, \zeta \,\rrbracket = M \in \mathcal{M} \\ \mathit{ff} & \text{otherwise} \end{cases} \\
\llbracket\, \mathit{is\_name}(\zeta) \,\rrbracket & = & \begin{cases} tt & \text{if } \zeta \in \mathcal{N} \\ \mathit{ff} & \text{otherwise} \end{cases} \\
\llbracket\, [\zeta = \eta] \,\rrbracket & = & \begin{cases} tt & \text{if } \zeta = \eta \in \mathcal{M} \\ \mathit{ff} & \text{otherwise} \end{cases}
\end{array}
$$

Table 2: Evaluation (decryption) in the spi calculus

Inspired by [DY83], (1) the environment learns new messages by reading any kind of data that the process sends on public channels; (2) the environment may then independent of any further message exchange update its knowledge by

- detecting whether a known message is a name or an encrypted message,
- using known names to decrypt known messages (usually called *analysis*),
- comparing known messages to other known messages,
- storing known messages for later comparison, encryption or decryption;

(3) the environment may then send on public channels any message that it is able to create (a procedure that is usually called *synthesis*) by using its current knowledge, plus fresh names that it might create itself. Summing up, in environments we need to represent knowledge in such a way that we may at any time calculate from this knowledge those messages that it can currently synthesize according to the above operations.

One straightforward approach is to jointly model the behavior of a pair $e_P \vdash P$, where e_P contains the current knowledge as the list of all messages ever received by the environment from the observed process that is now in state P. Any input action of P is governed by the environment in that we consider inputs for only those messages that can be synthesized by e_P. Any output action of P resulting in P' is used to increase the knowledge of e_P resulting in $e_{P'}$ by simply appending the new message to the list.

An alternative approach is to represent the knowledge "more efficiently" in *irreducible* form by carrying out the full analysis after every process output. An advantage is that the required data structure becomes smaller, and that the synthesis can be carried out directly. Note that we cannot represent the full synthesis, because it usually is infinite.

The mode of observation changes slightly under the regime of bisimulation. Instead of observing only a single process (as in $e_P \vdash P$), an environment now observes a pair of processes "at the same time" (as in $e_{PQ} \vdash P \mathrel{\mathcal{S}} Q$, where $\mathcal{S}$ denotes a bisimulation relation, and where e_{PQ} may be simply a pair (e_P, e_Q) of single environments). The spi calculus principle of distinguishing between *possessing* a message (e.g., $\mathrm{E}_k(a)$) and *knowing* it (e.g., knowing that k is the encryption key, such that a becomes known as well) comes into play again: while in standard process calculi it is required that the messages emitted by processes P and Q must coincide syntactically, the spi calculus must be more permissive. A common environment for the bisimulation game must permit different messages to be sent to the environment by P and Q, but only under the requirement that they lead to corresponding analyses in the respective environment component of e_{PQ}. This idea is captured by the notion of *consistency* which guarantees that an environment can not decrypt a message received from P unless it can also decrypt *the* corresponding message received from Q.

$$(\textsc{out})\ \frac{}{\overline{a}\langle M\rangle.P \xrightarrow{\overline{a}\,M} P} \qquad (\textsc{inp})\ \frac{}{a(x).P \xrightarrow{a\,M} P\{^{M}\!/_{x}\}} \qquad (\textsc{sum})\ \frac{P \xrightarrow{\mu} P'}{P + Q \xrightarrow{\mu} P'}$$

$$(\textsc{rep})\ \frac{P \,|\, !\,P \xrightarrow{\mu} P'}{!\,P \xrightarrow{\mu} P'} \qquad (\textsc{alp})\ \frac{Q \xrightarrow{\mu} Q' \quad P \equiv_{\alpha} Q}{P \xrightarrow{\mu} Q'} \qquad (\textsc{res})\ \frac{P \xrightarrow{\mu} P'}{(\nu c)\,P \xrightarrow{\mu} (\nu c)\,P'} \text{ if } c \notin \mathrm{n}(\mu)$$

$$(\textsc{open})\ \frac{P \xrightarrow{(\nu\tilde{b})\,\overline{a}\,M} P'}{(\nu c)\,P \xrightarrow{(\nu c\tilde{b})\,\overline{a}\,M} P'} \text{ if } \ \mathrm{n}(M) \ni c \notin \{a, \tilde{b}\} \qquad (\textsc{par})\ \frac{P \xrightarrow{\mu} P'}{P \,|\, Q \xrightarrow{\mu} P' \,|\, Q} \text{ if } \ \mathrm{bn}(\mu) \cap \mathrm{fn}(Q) = \emptyset$$

$$(\textsc{com})\ \frac{P \xrightarrow{a\,M} P' \quad Q \xrightarrow{(\nu\tilde{b})\,\overline{a}\,M} Q'}{P \,|\, Q \xrightarrow{\tau} (\nu\tilde{b})\,(P' \,|\, Q')} \text{ if } \{\tilde{b}\} \cap \mathrm{fn}(P) = \emptyset \qquad (\textsc{guard})\ \frac{P \xrightarrow{\mu} P'}{\phi P \xrightarrow{\mu} P'} \text{ if } [\![\, \phi \,]\!] = tt$$

$$(\textsc{let})\ \frac{P\{^{[\![\,\zeta\,]\!]}\!/_{z}\} \xrightarrow{\mu} P'}{\text{let } z = \zeta \text{ in } P \xrightarrow{\mu} P'} \text{ if } [\![\, \zeta \,]\!] \neq \perp$$

Table 3: Operational semantics of the spi calculus

As a consequence, environments must also properly keep track of the association of the messages received from P and Q.

Depending on the type of the data structure e introduced further below (frame-theory pairs, hedges, or substitution pairs) the notions of analysis, synthesis, and consistency appear in different forms or only implicitly, which renders their comparison non-trivial. Irreducibility may be enforced on the data structure. Synthesis may be expressed explicitly by means of substitution on expressions.

To relate two processes P and Q, one usually wants to find a bisimulation $\mathcal{S}$ such that $e \vdash P \;\mathcal{S}\; Q$ for some environment e that knows at least all the free names of both processes. Free names are public, so under a worst case assumption a malicious attacker might take advantage of any and all of them. Whenever $\mathcal{R} \subseteq \mathbf{E} \times \mathcal{P} \times \mathcal{P}$ is an environment-sensitive relation for some kind of environments $\mathbf{E}$, we define $\mathcal{R}^{-1} := \{\, (e^{-1}, Q, P) \mid (e, P, Q) \in \mathcal{R} \,\}$ for some suitably defined inversed environment e^{-1}. We write that $e \vdash P \;\mathcal{R}\; Q$ if $(e, P, Q) \in \mathcal{R}$, otherwise $e \not\vdash P \;\mathcal{R}\; Q$. $\mathcal{R}$ is *symmetric* if $\mathcal{R} = \mathcal{R}^{-1}$.

3.1 Framed and Fenced Bisimulations

Framed bisimulation, as introduced by [AG98], was the first environment-sensitive bisimulation proposed for the spi calculus. The original definition was for a strong and late bisimulation. Here, we study a weak and early variant in order to sharpen the comparison with the bisimulation defined in [BDP99, BDP02]. Abadi and Gordon also used a different calculus, with a complex set of messages containing integers, pairing and general encryption keys but without general guards, choice and general "let". However, the examples distinguishing the bisimularities, that we will exhibit in Section 4, can be expressed in the original spi calculus [AG99].

In framed bisimulation the environment consists of a frame and a theory. A frame is a set of names known to the environment. A theory is a set of pairs of messages considered equivalent by the environment.

Definition 3.1.1 *A* frame *is a finite subset of* $\mathcal{N}$*. A* theory *is a finite subset of* $\mathcal{M} \times \mathcal{M}$*.* **FT** *is the set of all frame-theory pairs. If* $B \subset \mathcal{M}$ *is finite then we define* $\mathrm{Id}_B := \{\, (b, b) \mid b \in B \,\}$*.*

If th *is a theory, we define* $\mathrm{th}^{-1} := \{\, (N, M) \mid (M, N) \in \mathrm{th} \,\}$, $\pi_1(\mathrm{th}) := \{\, M \mid (M, N) \in \mathrm{th} \,\}$ *and* $\pi_2(\mathrm{th}) := \{\, N \mid (M, N) \in \mathrm{th} \,\}$. *The names of a theory is defined as* $\mathrm{n}(\mathrm{th}) := \mathrm{n}(\pi_1(\mathrm{th}) \cup \pi_2(\mathrm{th}))$.

A frame-theory pair is *consistent* if the theory only contains pairs of encrypted messages that the environment can not decrypt. The environment can not consider a given message equivalent to two different messages.

Definition 3.1.2 *A frame-theory pair* $(\mathrm{fr}, \mathrm{th})$ *is* consistent *iff for all messages* M *and* N *such that* $(M, N) \in \mathrm{th}$ *we have that*

1. $M, N \notin \mathcal{N}$
2. *If* $(M', N') \in \mathrm{th}$ *then* $M = M' \iff N = N'$
3. *If* $M = \mathrm{E}_a(M')$ *and* $N = \mathrm{E}_b(N')$ *then* $\mathrm{fr} \cap \{a, b\} = \emptyset$.

Example 3.1.3 *One consistent and three inconsistent frame-theory pairs:*

$$(\{a, b\}, \{(\mathrm{E}_k(a), \mathrm{E}_k(a)), (\mathrm{E}_l(a), \mathrm{E}_l(b))\}) \text{ is consistent.}$$
$$(\mathrm{fr}, \mathrm{th}) = (\{a, b\}, \{(\mathrm{E}_k(a), k)\}) \text{ violates condition 1 for consistency.}$$
$$(\mathrm{fr}', \mathrm{th}') = (\{a, b\}, \{(\mathrm{E}_k(a), \mathrm{E}_b(a))\}) \text{ violates condition 3 for consistency.}$$
$$(\mathrm{fr} \cup \mathrm{fr}', \mathrm{th} \cup \mathrm{th}') = (\{a, b\}, \{(\mathrm{E}_k(a), k), (\mathrm{E}_k(a), \mathrm{E}_b(a))\}) \text{ violates all three conditions for consistency.}$$

The *synthesis* $\mathcal{S}(\cdot)$ of a frame-theory pair is the set of message pairs constructed by encrypting message pairs from the theory with keys from the frame. The environment considers equivalent any message pair in the synthesis.

Definition 3.1.4 *If* $(\mathrm{fr}, \mathrm{th})$ *is a frame-theory pair, we let* $\mathcal{S}(\mathrm{fr}, \mathrm{th})$ *be the smallest subset of* $\mathcal{M} \times \mathcal{M}$ *containing* $\mathrm{th} \cup \mathrm{Id}_{\mathrm{fr}}$ *and satisfying*

$$(\textsc{syn-enc})\ \frac{(M, N) \in \mathcal{S}(\mathrm{fr}, \mathrm{th}) \qquad (a, b) \in \mathcal{S}(\mathrm{fr}, \mathrm{th})}{(\mathrm{E}_a(M), \mathrm{E}_b(N)) \in \mathcal{S}(\mathrm{fr}, \mathrm{th})}$$

We write $(\mathrm{fr}, \mathrm{th}) \vdash M \leftrightarrow N$ *for* $(M, N) \in \mathcal{S}(\mathrm{fr}, \mathrm{th})$; *otherwise* $(\mathrm{fr}, \mathrm{th}) \not\vdash M \leftrightarrow N$.

To compare the knowledge of environments we use the following pre-order:

Definition 3.1.5 $(\mathrm{fr}, \mathrm{th}) \le (\mathrm{fr}', \mathrm{th}')$ *iff* $\mathcal{S}(\mathrm{fr}, \mathrm{th}) \subseteq \mathcal{S}(\mathrm{fr}', \mathrm{th}')$. *Two frame-theory pairs* $(\mathrm{fr}, \mathrm{th})$ *and* $(\mathrm{fr}', \mathrm{th}')$ *are* $\mathcal{M}$-equivalent, *written* $(\mathrm{fr}, \mathrm{th}) \gtreqless (\mathrm{fr}', \mathrm{th}')$ *when* $\mathcal{S}(\mathrm{fr}, \mathrm{th}) = \mathcal{S}(\mathrm{fr}', \mathrm{th}')$.

A *framed process pair* is a triple $((\mathrm{fr}, \mathrm{th}), P, Q)$ where fr is a frame, th is a theory and P and Q are processes. A *framed relation* $\mathcal{R}$ is a set of framed process pairs. $\mathcal{R}$ is *consistent* if $(\mathrm{fr}, \mathrm{th})$ is consistent whenever $(\mathrm{fr}, \mathrm{th}) \vdash P \mathrel{\mathcal{R}} Q$. Now we have got enough notation to define framed bisimilarity.

Definition 3.1.6 *A consistent framed relation* $\mathcal{R}$ *is a* framed simulation *if whenever* $(\mathrm{fr}, \mathrm{th}) \vdash P \mathrel{\mathcal{R}} Q$ *we have that*

1. *If* $P \xrightarrow{\tau} P'$ *then there exists* Q' *such that* $Q \Longrightarrow Q'$ *and* $(\mathrm{fr}, \mathrm{th}) \vdash P' \mathrel{\mathcal{R}} Q'$.
2. *If* $P \xrightarrow{a\,M} P'$, $a \in \mathrm{fr}$, $B \subset \mathcal{N}$ *is finite,* $B \cap (\mathrm{fn}(P, Q) \cup \mathrm{fr} \cup \mathrm{n}(\mathrm{th})) = \emptyset$, $N \in \mathcal{M}$, *and* $(\mathrm{fr} \cup B, \mathrm{th}) \vdash M \leftrightarrow N$,
   then there exists Q'
   such that $Q \xRightarrow{a\,N} Q'$ *and* $(\mathrm{fr} \cup B, \mathrm{th}) \vdash P' \mathrel{\mathcal{R}} Q'$
3. *If* $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $a \in \mathrm{fr}$, *and* $\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{fr} \cup \mathrm{n}(\pi_1(\mathrm{th}))) = \emptyset$,
   then there exist $Q', N, \tilde{d}$ *with* $\{\tilde{d}\} \cap (\mathrm{fn}(Q) \cup \mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th}))) = \emptyset$ *and*

   (a) $Q \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} Q'$, *and*

```
$\xi(\mathrm{fr}, \mathrm{th}, M, N)$
    IF $((\mathrm{fr}, \mathrm{th}) \vdash M \leftrightarrow N)$ THEN RETURN $(\mathrm{fr}, \mathrm{th})$
    IF $(M = N \in \mathcal{N})$ DO
        $(\mathrm{fr}', \mathrm{th}') := (\mathrm{fr} \cup \{M\}, \mathrm{th})$
        $\lambda := \emptyset$
        FOR EACH $(\mathrm{E}_k(M'), \mathrm{E}_l(N')) \in \mathrm{th}$ DO
            IF $(k = l = M)$ THEN DO
                $\mathrm{th}' := \mathrm{th}' \setminus \{(\mathrm{E}_k(M'), \mathrm{E}_l(N'))\}$
                $\lambda := \lambda \cup \{(M', N')\}$
            ELSIF $(k = M \vee l = M)$ THEN RETURN $\perp$
            DONE
        DONE
        FOR EACH $(M', N') \in \lambda$ DO
            $(\mathrm{fr}', \mathrm{th}') := \xi(\mathrm{fr}', \mathrm{th}', M', N')$
        DONE
        RETURN $(\mathrm{fr}', \mathrm{th}')$
    ELSIF $(M = \mathrm{E}_k(M') \wedge N = \mathrm{E}_l(N'))$ DO
        IF $(k = l \in \mathrm{fr})$ THEN RETURN $\xi(\mathrm{fr}, \mathrm{th}, M', N')$
        IF $(k \in \mathrm{fr} \vee l \in \mathrm{fr})$ THEN RETURN $\perp$
        RETURN $(\mathrm{fr}, \mathrm{th} \cup \{(M, N)\})$
    DONE
    RETURN $\perp$
DONE
```

Table 4: An algorithm for the function ξ

(b) *there exist* $\mathrm{fr}', \mathrm{th}'$ *with* $\mathcal{S}(\mathrm{fr}, \mathrm{th}) \subseteq \mathcal{S}(\mathrm{fr}', \mathrm{th}')$ *and* $(\mathrm{fr}', \mathrm{th}') \vdash M \leftrightarrow N$ *such that* $(\mathrm{fr}', \mathrm{th}') \vdash P' \mathrel{\mathcal{R}} Q'$.

$\mathcal{R}$ *is a* framed bisimulation *if both* $\mathcal{R}$ *and* $\mathcal{R}^{-1}$ *are framed simulations.*

It is worth noting how the new environment after output transitions is characterized: names and message pairs can be freely added as long as the synthesis is extended conservatively and the new output messages are kept indistinguishable.
Since any union of framed bisimulations is a framed bisimulation there exists a greatest framed bisimulation, denoted $\approx_{\mathrm{f}}$, which is the union of all framed bisimulations.

Fenced bisimulation was defined by [EHHO99], who proved it to be a sound and complete approximation to framed bisimulation. (In Section 4.1, we show that this is in fact not the case.) The difference between the definitions is that fenced bisimulation replaces the existential quantification over frames and theories in case *3.(b)* of Definition 3.1.6 with a function ξ, defined in Table 4, that extends a given frame-theory pair with a new pair of messages. Elkjær et al. showed that ξ creates a *minimal* consistent extension whenever there exists one.

Since our message grammar is simpler the definition of ξ can be simplified for our case.

Definition 3.1.7 *A consistent framed relation* $\mathcal{R}$ *is a* fenced simulation *if whenever* $(\mathrm{fr}, \mathrm{th}) \vdash P \mathrel{\mathcal{R}} Q$ *we have that*

1. *If* $P \xrightarrow{\tau} P'$
   then there exists Q' *such that*
   $Q \Longrightarrow Q'$ *and* $(\mathrm{fr}, \mathrm{th}) \vdash P' \mathrel{\mathcal{R}} Q'$.
2. *If* $P \xrightarrow{a\,M} P'$, $a \in \mathrm{fr}$, $B \subset \mathcal{N}$ *is finite,* $B \cap (\mathrm{fn}(P, Q) \cup \mathrm{fr} \cup \mathrm{n}(\mathrm{th})) = \emptyset$, $N \in \mathcal{M}$, *and* $(\mathrm{fr} \cup B, \mathrm{th}) \vdash M \leftrightarrow N$,
   then there exists Q' *such that*
   $Q \xRightarrow{a\,N} Q'$ *and* $(\mathrm{fr} \cup B, \mathrm{th}) \vdash P' \mathrel{\mathcal{R}} Q'$

3. If $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $a \in \mathrm{fr}$ *and* $\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{fr} \cup \mathrm{n}(\pi_1(\mathrm{th}))) = \emptyset$
there exist $Q', N, \tilde{d}$ *with* $\{\tilde{d}\} \cap (\mathrm{fn}(Q) \cup \mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th}))) = \emptyset$
such that $Q \overset{(\nu\tilde{d})\,\overline{a}\,N}{\Longrightarrow} Q'$ *and* $\xi(\mathrm{fr}, \mathrm{th}, M, N) \vdash P' \,\mathcal{R}\, Q'$.

$\mathcal{R}$ *is a* fenced bisimulation *if both* $\mathcal{R}$ *and* $\mathcal{R}^{-1}$ *are fenced simulations.*

Since any union of fenced bisimulations is a fenced bisimulation there exists a greatest fenced bisimulation, denoted $\approx_{\#}$, which is the union of all fenced bisimulations.

3.2 Alley and Trellis Bisimulations

Boreale et al. [BDP99, BDP02] define environment-sensitive semantics and a corresponding weak bisimulation for the spi calculus. As mentioned earlier, their bisimulation is called *alley* in this paper in order to distinguish it from the other bisimulations, which are also environment-sensitive. The authors also prove that alley bisimulation is a sound approximation of barbed equivalence, and that the approximation is complete for the class of "structurally image-finite" processes. They also study a number of "up-to" techniques for this bisimulation.

Formally, Boreale et al. define two levels of operational semantics, one for the behavior of processes, and another one for the corresponding behavior of environments. Following [FHJ01], we adapt this formalization to the style where the environment behavior is built into the definition of bisimulation.

In alley bisimulation, the environment is a pair of substitutions.

Definition 3.2.1 *A* substitution σ *is a finite partial function* $\mathcal{N} \to \mathcal{M}$. *We write* $\sigma\{^{M}/_{x}\}$ *for* $\sigma \cup \{(x, M)\}$, *where* $x \notin \mathrm{dom}(\sigma)$. *To add several messages, we write* $\sigma\{^{M_1}/_{x_1}, \ldots, ^{M_n}/_{x_n}\}$ *for* $\sigma \cup \{\, (x_i, M_i) \mid i = 1, 2, \ldots, n \,\}$ *where the* x_i *are assumed to be pairwise different and not in* $\mathrm{dom}(\sigma)$. *The set of free names of a substitution is defined as* $\mathrm{fn}(\sigma) := \mathrm{n}(\mathrm{range}(\sigma))$. *We denote by* **SS** *the set of all alleys, i.e., the set of all substitution pairs.*

Any set of messages, e.g., the messages in the codomain of a substitution, might be reduced via decryption using the notion of analysis.

Definition 3.2.2 *The* analysis $\mathcal{A}(S)$ *and the* irreducibles $\mathcal{I}(S)$ *of a set* $S \subseteq \mathcal{M}$ *are defined as follows:* $\mathcal{A}(S)$ *is the smallest subset of* $\mathcal{M}$ *containing* S *and satisfying*

$$(\textsc{set-dec})\ \frac{\mathrm{E}_a(M) \in \mathcal{A}(S) \qquad a \in \mathcal{A}(S)}{M \in \mathcal{A}(S)}$$

and $\mathcal{I}(S) := \mathcal{A}(S) \setminus \{\, \mathrm{E}_a(M) \mid a \in \mathcal{A}(S) \,\}$.

We introduce a function that decrypts messages M as far as possible, i.e., peels out the *core* of M using the knowledge of a substitution σ. We use the shorthands $\mathcal{I}(\sigma)$ for $\mathcal{I}(\mathrm{range}(\sigma))$ and $\mathcal{A}(\sigma)$ for $\mathcal{A}(\mathrm{range}(\sigma))$. We define

$$\mathrm{core}_\sigma(M) \overset{\mathrm{def}}{=} \begin{cases} \mathrm{core}_\sigma(M') & \text{if } M = \mathrm{E}_a(M') \text{ and } a \in \mathcal{I}(\sigma) \\ M & \text{otherwise} \end{cases}$$

Thus, we can decompose any message M into $\mathrm{E}_{b_n}(\cdots \mathrm{E}_{b_2}(\mathrm{E}_{b_1}(\mathrm{core}_\sigma(M)))\cdots)$ for any substitution σ with $\{b_1 \ldots, b_n\} \subseteq \mathcal{I}(\sigma)$; if $\mathrm{core}_\sigma(M) = \mathrm{E}_a(N)$, then $a \notin \mathcal{I}(\sigma)$. As a special case, we define $\mathcal{C}(\sigma, x) := \mathrm{core}_\sigma(\sigma(x))$. Therefore, $\mathcal{I}(\sigma) = \{\mathcal{C}(\sigma, x) \mid x \in \mathrm{dom}(\sigma)\}$.

Two substitutions are consistent if they decrypt corresponding messages in precisely corresponding ways. In other words, it does not suffice to just decrypt to corresponding cores, but we also must use corresponding keys for the decryption.

Definition 3.2.3 *A pair of substitutions* (σ, ρ) *is* consistent, *written* $\sigma \sim \rho$, *iff* σ *and* ρ *have the same domain* $\{x_1 \ldots, x_n\}$ *and the following conditions hold:*

1. $\mathcal{C}(\sigma, x_i) \in \mathcal{N} \iff \mathcal{C}(\rho, x_i) \in \mathcal{N}$
2. $\mathcal{C}(\sigma, x_i) = \mathcal{C}(\sigma, x_j) \iff \mathcal{C}(\rho, x_i) = \mathcal{C}(\rho, x_j)$
3. *For each* $i = 1, 2, \ldots, n$ *there is a tuple* $\tilde{\iota} = \iota_1..\iota_n$ *such that*

$$\begin{array}{rcl} \sigma(x_i) & = & \mathrm{E}_{\mathcal{C}(\sigma,x_{\iota_m})}(\cdots \mathrm{E}_{\mathcal{C}(\sigma,x_{\iota_2})}(\mathrm{E}_{\mathcal{C}(\sigma,x_{\iota_1})}(\mathcal{C}(\sigma, x_i))) \cdots) \\ \rho(x_i) & = & \mathrm{E}_{\mathcal{C}(\rho,x_{\iota_m})}(\cdots \mathrm{E}_{\mathcal{C}(\rho,x_{\iota_2})}(\mathrm{E}_{\mathcal{C}(\rho,x_{\iota_1})}(\mathcal{C}(\rho, x_i))) \cdots) \end{array}$$

Example 3.2.4 *To give some examples on this definition, we let*

$$\begin{aligned} \sigma &= \{{}^{a}/_{x_1}\}\{{}^{\mathrm{E}_a(b)}/_{x_2}\}\{{}^{\mathrm{E}_k(\mathrm{E}_a(c))}/_{x_3}\} \\ \rho &= \{{}^{a}/_{x_1}\}\{{}^{\mathrm{E}_a(b)}/_{x_2}\}\{{}^{\mathrm{E}_k(\mathrm{E}_k(d))}/_{x_3}\} \\ \tau &= \{{}^{a}/_{x_1}\}\{{}^{\mathrm{E}_a(c)}/_{x_2}\}\{{}^{\mathrm{E}_k(\mathrm{E}_k(c))}/_{x_3}\} \end{aligned}$$

Then we have that $\sigma \sim \rho$, $\sigma \sim \tau$ *and* $\rho \sim \tau$. *Note that we must allow two different names (b and c) to correspond, in order to relate* σ *and* τ. *If both* σ *and* ρ *acquire knowledge of the key (name) k we get that* $\sigma\{{}^{k}/_{y}\} \not\sim \rho\{{}^{k}/_{y}\}$, *since they violate condition 3 of* $\sim$ *by using different encryption keys in the decryption of the third message. We also have that* $\rho\{{}^{k}/_{y}\} \not\sim \tau\{{}^{k}/_{y}\}$, *since they violate condition 2 by having c in* $\mathcal{I}(\tau)$ *correspond to both b and d in* $\mathcal{I}(\rho)$.

We also define a notion of the synthesis of a consistent substitution pair.

Definition 3.2.5 *If* $\sigma \sim \rho$ *we write* $(\sigma, \rho) \vdash M \leftrightarrow N$ *iff there is* ζ *such that* $\mathrm{n}(\zeta) \subseteq \mathrm{dom}(\sigma)$, $[\![\, \zeta\sigma \,]\!] = M$ *and* $[\![\, \zeta\rho \,]\!] = N$. *The synthesis of a consistent pair of substitutions is defined as* $\mathcal{S}(\sigma, \rho) := \{(M, N) \mid (\sigma, \rho) \vdash M \leftrightarrow N\}$.

An alley process pair is a triple $((\sigma, \rho), P, Q)$ with $\mathrm{dom}(\sigma) = \mathrm{dom}(\rho)$. An *alley relation* $\mathcal{R}$ is a set of alley process pairs. $\mathcal{R}$ is *consistent* if $(\sigma, \rho) \vdash P \mathrel{\mathcal{R}} Q$ implies that $\sigma \sim \rho$.

Definition 3.2.6 *A consistent alley relation* $\mathcal{R}$ *is an* alley simulation *if whenever* $(\sigma, \rho) \vdash P \mathrel{\mathcal{R}} Q$ *the following conditions hold:*

1. *If* $P \xrightarrow{\tau} P'$ *then there exists* Q' *such that* $Q \Longrightarrow Q'$ *and* $(\sigma, \rho) \vdash P' \mathrel{\mathcal{R}} Q'$.
2. *If* $P \xrightarrow{a\,M} P'$ *and there are* $\zeta, \tilde{b}, b$
such that $[\![\, \zeta\sigma \,]\!] = M$ *with* $(\sigma, \rho) \vdash a \leftrightarrow b$, $\tilde{b} = \mathrm{n}(\zeta) \setminus \mathrm{dom}(\sigma)$ *and* $\tilde{b} \cap \mathrm{fn}(P, Q, \rho, \sigma) = \emptyset$, *then there exist* $\tilde{c}, Q'$ *with* $\tilde{c} \subset \mathcal{N}$, $|\tilde{c}| = |\tilde{b}|$, $\tilde{c} \cap \mathrm{dom}(\sigma) = \emptyset$
such that $Q \xRightarrow{b\,[\![\, \zeta\rho \,]\!]} Q'$ *and* $(\sigma\{{}^{\tilde{b}}/_{\tilde{c}}\}, \rho\{{}^{\tilde{b}}/_{\tilde{c}}\}) \vdash P' \mathrel{\mathcal{R}} Q'$.
3. *If* $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$ *with* $\mathrm{fn}(P, \sigma) \cap \{\tilde{c}\} = \emptyset$ *and* $(\sigma, \rho) \vdash a \leftrightarrow b$
then there are $Q', N, \tilde{d}, x$ *with* $\mathrm{fn}(Q, \rho) \cap \{\tilde{d}\} = \emptyset$
such that $Q \xRightarrow{(\nu\tilde{d})\,\overline{b}\,N} Q'$ *and* $(\sigma\{{}^{M}/_{x}\}, \rho\{{}^{N}/_{x}\}) \vdash P' \mathrel{\mathcal{R}} Q'$.

$\mathcal{R}$ *is an* alley bisimulation *if both* $\mathcal{R}$ *and* $\mathcal{R}^{-1}$ *are alley simulations.*

Note the difference with respect to the previous bisimulations. Here, the environment is extended simply by mechanically adding the new messages (for output) or the new names (for input), without reducing the environment at all. This also gives a minimal extension (cf. fenced bisimulation) since no extra information may be added. Since we use substitutions as environments, consistency is vital. Otherwise, the creation of message pairs by applying both substitutions to the same formula gives meaningless results.

Trellis bisimulation is a strengthened variant of alley bisimulation, studied (not under this name) by [BDP99]. There, two different notions of consistency of environments were proposed, of which one was rejected since it was considered too strong. It constitutes the basis for trellis bisimulation.

Definition 3.2.7 *A consistent pair of substitutions* $\sigma \sim \rho$ *is strongly consistent, written* $\sigma \sim_{\mathrm{s}} \rho$, *if* $\mathcal{C}(\sigma, x) \in \mathcal{N}$ *implies that* $\mathcal{C}(\sigma, x) = \mathcal{C}(\rho, x)$.

This resembles the definition of consistency of frame-theory pairs in that two different names may never be considered equal. Strong consistency has a corresponding bisimulation, called *trellis* in this paper, that was defined and compared to fenced bisimulation by [FHJ01]. We recapitulate and strengthen the results of the comparison in Section 7.

Definition 3.2.8 *An alley relation $\mathcal{R}$ is strongly consistent if $(\sigma, \rho) \vdash P \mathrel{\mathcal{R}} Q$ implies $\sigma \sim_{\mathrm{s}} \rho$. We call* trellis bisimulation *a strongly consistent alley bisimulation.*

Since any union of alley/trellis bisimulations is an alley/trellis bisimulation itself there exists a greatest alley/trellis bisimulation, denoted $\approx_{\mathrm{a}}/\approx_{\mathrm{s}}$, which is the union of all alley/trellis bisimulations.

3.3 Hedged Bisimulation

Hedged bisimulation is introduced in this paper in order to clarify the differences between framed, fenced, and alley bisimulation. Recall that alley bisimulation, unlike its counterparts, does not force two processes to always send the same names; it rather remembers that the respective names correspond to each other. The basic idea of hedges is to mimic the lack of correspondence in frame-theory pairs by dropping the separate frame component, but to include corresponding names as part of the theory. The resulting theory is then called a *hedge*.

Definition 3.3.1 *A* hedge *is a theory, i.e., a finite subset of $\mathcal{M}^2$. We denote by $\mathbf{H}$ the set of all hedges. The* synthesis *$\mathcal{S}(\cdot)$ of a hedge is defined as $\mathcal{S}(h) = \mathcal{S}(\emptyset, h)$. We write $h \vdash M \leftrightarrow N$ for $(M, N) \in \mathcal{S}(h)$, $h \nvdash M \leftrightarrow N$ otherwise.*

A hedge is consistent if the hedge only contains pairs of names and pairs of encrypted messages that can not be decrypted by the environment. We also require that no message is considered to be equivalent to two different messages.

Definition 3.3.2 *A hedge h is* consistent *iff whenever $(M, N) \in h$*

1. *$M \in \mathcal{N} \iff N \in \mathcal{N}$*
2. *If $(M', N') \in h$ then $M = M' \iff N = N'$*
3. *If $M = \mathrm{E}_a(M')$ and $N = \mathrm{E}_b(N')$ then $a \notin \pi_1(h)$ and $b \notin \pi_2(h)$.*

Example 3.3.3 *A consistent hedge and four inconsistent hedges:*

$$\begin{aligned}
g &= \{(a,a), (b,c), (c,k), (\mathrm{E}_k(b), \mathrm{E}_l(a))\} \text{ is consistent.}\\
h_1 &= \{(a, \mathrm{E}_k(a)), (b,c), (\mathrm{E}_k(b), \mathrm{E}_l(a))\} \text{ violates condition 1 for consistency.}\\
h_2 &= \{(a,c), (b,c), (\mathrm{E}_k(b), \mathrm{E}_l(a))\} \text{ violates condition 2 for consistency.}\\
h_3 &= \{(a,a), (k,c), (\mathrm{E}_k(b), \mathrm{E}_l(a))\} \text{ violates condition 3 for consistency.}
\end{aligned}$$

Note that $h_1 \cup h_2 \cup h_3$ violates all three conditions. Indeed, if a hedge h is not consistent then $h \cup h'$ is not consistent for any hedge h'.

The difference between a consistent hedge and a consistent frame-theory pair is that we do not require that the hedge receives the same names from both processes, so they do not need to use the same channels and encryption keys. The difference between a consistent hedge and a consistent substitution pair is that the former only contains undecryptable messages (i.e., cores) and that no duplicate message pairs are allowed. The third condition for consistent substitutions (cf. Definition 3.2.3) roughly corresponds to the definition of hedge analysis (cf. Section 6.4) .

Definition 3.3.4 *The* analysis *$\mathcal{A}(h)$ and the* irreducibles *$\mathcal{I}(h)$ of a hedge h are defined as follows: $\mathcal{A}(h)$ is the smallest subset of $\mathcal{M}^2$ containing h and satisfying*

$$(\textsc{hedge-dec})\ \frac{(\mathrm{E}_a(M), \mathrm{E}_b(N)) \in \mathcal{A}(h) \qquad (a,b) \in \mathcal{A}(h)}{(M,N) \in \mathcal{A}(h)}$$

and $\mathcal{I}(h) \overset{\mathrm{def}}{=} \mathcal{A}(h) \setminus \{\, (\mathrm{E}_a(M), \mathrm{E}_b(N)) \mid (a,b) \in \mathcal{A}(h) \wedge M, N \in \mathcal{M} \,\}$.

The analysis of hedges decrypts pairs of messages using pairs of names that are considered equivalent by the environment. The resulting notion of irreducibles in fact also corresponds to the ξ-function of fenced bisimulation.

Now that the environment and notions of consistency are defined, the definition of hedged bisimulation is straightforward. A *hedged relation* $\mathcal{R}$ is a subset of $\mathbf{H} \times \mathcal{P} \times \mathcal{P}$. We say that $\mathcal{R}$ is *consistent* if $h \vdash P \mathrel{\mathcal{R}} Q$ implies that h is consistent.

Definition 3.3.5 *A consistent hedged relation $\mathcal{R}$ is a* hedged simulation *if whenever $h \vdash P \mathrel{\mathcal{R}} Q$ we have that*

1. *If $P \xrightarrow{\tau} P'$ then there exists Q' such that $Q \Longrightarrow Q'$ and $h \vdash P' \mathrel{\mathcal{R}} Q'$.*
2. *If $P \xrightarrow{a\,M} P'$, $h \vdash a \leftrightarrow b$, $B \subset \mathcal{N}$ is finite, $B \cap (\mathrm{fn}(P,Q) \cup \mathrm{n}(h)) = \emptyset$, $N \in \mathcal{M}$, and $h \cup \mathrm{Id}_B \vdash M \leftrightarrow N$, then there exists Q' such that $Q \xRightarrow{b\,N} Q'$ and $h \cup \mathrm{Id}_B \vdash P' \mathrel{\mathcal{R}} Q'$.*
3. *If $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $h \vdash a \leftrightarrow b$ and $\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{n}(\pi_1(h))) = \emptyset$ there exist $Q', N, \tilde{d}$ with $\{\tilde{d}\} \cap (\mathrm{fn}(Q) \cup \mathrm{n}(\pi_2(h))) = \emptyset$ such that $Q \xRightarrow{(\nu\tilde{d})\,\overline{b}\,N} Q'$ and $\mathcal{I}(h \cup \{(M,N)\}) \vdash P' \mathrel{\mathcal{R}} Q'$.*

$\mathcal{R}$ is a hedged bisimulation *if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are hedged simulations.*

On process output we use $\mathcal{I}(\cdot)$ to construct the new hedge after the transition. This entails applying all decryptions that the environment can do, producing—as in fenced bisimulation—the minimal extension of the hedge h with (M,N). As in the other bisimilations, this extension may turn out to be inconsistent, signifying that the hedge has detected a difference between the messages received from the process pair.

Since any union of hedged bisimulations is a hedged bisimulation there exists a greatest hedged bisimulation, denoted $\approx_{\mathrm{h}}$, which is the union of all hedged bisimulations.

4 Distinguishing Examples

In this section we exhibit differences between framed, fenced and hedged bisimulation. In Section 7 we show that hedged and alley bisimulation are equivalent, so these examples also distinguish alley bisimulation from framed and fenced.

4.1 Fenced vs. Framed/Hedged — Fresh Names Should be Just Fresh

The following example distinguishes fenced bisimulation from its competitors due to a subtle requirement concerning the data involved in the simulation of output transitions, especially the conditions on the choice of bound names:

$$\begin{array}{rclcrcl} P & := & (\nu nkl) \quad \overline{a}\langle \mathrm{E}_l(\mathrm{E}_k(n))\rangle.P' & \qquad & P' & := & (\nu m)\,\overline{a}\langle m\rangle.\,\mathbf{0} \\ Q & := & (\nu nk) \quad \overline{a}\langle \mathrm{E}_k(n)\rangle.Q' & & Q' & := & (\nu m)\,\overline{a}\langle m\rangle.\,\mathbf{0} \end{array}$$

Although there is no reason for P and Q to be distinguished, fenced bisimulation does so because it insists that single fresh names be simulated without renaming.

We write $\mathrm{pwd}(\tilde{n})$ to denote that $\tilde{n}$ is a tuple of *pair*w*ise* d*ifferent names.

Proposition 4.1.1 *$(\{a\},\emptyset) \not\vdash P \approx_{\#} Q$.*

Proof. The transitions of P are of the form $P \xrightarrow{(\nu nkl)\,\overline{a}\langle \mathrm{E}_l(\mathrm{E}_k(n))\rangle.} P'$ where $\mathrm{pwd}(n,k,l,a)$. The transitions of Q are of the form $Q \xrightarrow{(\nu n'k')\,\overline{a}\langle \mathrm{E}_{k'}(n')\rangle.} Q'$ where $\mathrm{pwd}(n',k',a)$. We then get the theory $\{(\mathrm{E}_l(\mathrm{E}_k(n)), \mathrm{E}_{k'}(n'))\}$. Since n,k,l are pairwise different, there must be a name $z \in \{n,k,l\} \setminus \{n',k'\}$.

Now P' must simulate the transition $Q' \xrightarrow{(\nu z)\,\overline{a}\,z} \mathbf{0}$. We have $P' \xrightarrow{(\nu m)\,\overline{a}\,m} \mathbf{0}$ for all $m \neq a$. For the resulting frame-theory pair to be consistent, we must have $m = z$, but since $z \in \mathrm{n}(\pi_1(\mathrm{th}))$ this transition can not be used. □

Fenced bisimulation fails since it cannot simulate an output of a particular bound name when this name is already known by the simulating environment. Here, both framed and hedged bisimulation succeed, but in different ways.

Proposition 4.1.2 $(\{a\}, \emptyset) \vdash P \approx_{\mathrm{f}} Q$.

Proof. A framed bisimulation relating P and Q is given by

$$\begin{aligned}\mathcal{R} := \{&((\{a\},\emptyset), P, Q)\}\\ \cup\{&((\{a,k\},\{(\mathrm{E}_l(\mathrm{E}_k(n)),\mathrm{E}_l(n))\}), P', Q') \mid \mathrm{pwd}(a,k,l,n)\}\\ \cup\{&((\{a,k,m\},\{(\mathrm{E}_l(\mathrm{E}_k(n)),\mathrm{E}_l(n))\}), \mathbf{0}, \mathbf{0}) \mid \mathrm{pwd}(a,k,l,m,n)\}\end{aligned}$$

Note the addition of k to the frame, which relieves the process P' from simulating the critical bound output of z by Q' (see the previous proof). The name created by Q' must be different from the names in the current frame and theory, so by simply adding k to the frame—which is allowed in framed bisimulation, but due to the minimality property not in fenced bisimulation—this name must be chosen different from k, and thus P' may also create it. □

Proposition 4.1.3 $\{(a,a)\} \vdash P \approx_{\mathrm{h}} Q$.

Proof. A hedged bisimulation relating P and Q is given by

$$\begin{array}{rcl}\mathcal{R} & := & \{\ (\{(a,a)\}, P, Q)\ \}\\ & \cup & \{\ (\{(a,a), (\mathrm{E}_l(\mathrm{E}_k(n)),\mathrm{E}_l(n))\}, P', Q') \mid \mathrm{pwd}(a,k,l,n)\ \}\\ & \cup & \{\ (\{(a,a), (\mathrm{E}_l(\mathrm{E}_k(n)),\mathrm{E}_l(n)), (m,m)\}, \mathbf{0}, \mathbf{0}) \mid \mathrm{pwd}(a,k,l,n,m)\ \}\\ & \cup & \{\ (\{(a,a), (\mathrm{E}_l(\mathrm{E}_k(n)),\mathrm{E}_l(n)), (w,k)\}, \mathbf{0}, \mathbf{0}) \mid \mathrm{pwd}(a,k,l,n,w)\ \}\end{array}$$

Note the addition of (w,k) to the hedge, denoting that the environment cannot distinguish the two different names. □

In comparison, we may conclude that $\approx_{\mathrm{f}}$ equates the processes through a non-minimal extension of the frame (which could be considered “cheating”), while $\approx_{\mathrm{h}}$ equates them (more adequately) through correspondence.

Corollary 4.1.4 $\approx_{\mathrm{f}}$ *is not a subset of* $\approx_{\#}$.

An Up-To Interpretation.

The example above shows that deleting information (the name k) from a frame-theory pair actually may increase its power to distinguish between processes! [BDP02] call this removal of knowledge from the environment *weakening*[1], which should intuitively be sound with respect to a bisimulation: An environment with more information at hand has more possibilities to discover a difference between two processes. While Theorem 7.4.4 will state that hedged bisimulation is sound up to weakening, the above examples show that this is not the case for framed and fenced bisimulation.

Proposition 4.1.5 *Framed and fenced bisimulations are not sound up to weakening.*

[1] This is in contrast to the field of type systems, where the notion of weakening usually denotes the *addition* (!) of potentially unnecessary information to the (typing) environment, while the typability judgment is preserved. Here, for environment-sensitive relations in the spirit of [BDP02], “up to weakening permits discarding environment entries”, thus it somehow denotes the *removal* (!) of environment knowledge.

4.2 Framed vs. Hedged — Unknown Names Must Not Matter

This example is a version without pairing of the example used by [AG98] to conjecture that framed bisimilarity is not complete w.r.t. barbed equivalence. We define

$$\begin{array}{lcll} P & := & (\nu klm) & \overline{a}\langle \mathrm{E}_k(\mathrm{E}_l(m))\rangle.\ \ (\ \ \overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0}\ \ ) \\ Q & := & (\nu km) & \overline{a}\langle \mathrm{E}_k(m)\rangle.\ \qquad\quad \overline{a}\langle m\rangle.\mathbf{0} \end{array}$$

We show that $\{(a,a)\} \vdash P \approx_{\mathrm{h}} Q$ and that $(\{a\},\emptyset) \not\vdash P \approx_{\mathrm{f}} Q$. Intuitively, this means that for $\approx_{\mathrm{f}}$ the identity of the unknown name matters, although an attacker doesn't have any means to verify this identity.

Proposition 4.2.1 $\{(a,a)\} \vdash P \approx_{\mathrm{h}} Q$.

Proof. We show that the relation

$$\begin{aligned} \mathcal{R} :=& \{\ (\{(a,a)\}, P, Q)\ \} \\ &\cup\{\ (h(k,l,m), (\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0}), \overline{a}\langle m\rangle.\mathbf{0}) \mid k,l,m \in \mathcal{N}\setminus\{a\}\ \} \\ &\cup\{\ (h(k,l,m) \cup \{(m,m)\}, \mathbf{0}, \mathbf{0}) \mid k,l,m \in \mathcal{N}\setminus\{a\}\ \} \\ &\cup\{\ (h(k,l,m) \cup \{(l,m)\}, \mathbf{0}, \mathbf{0}) \mid k,l,m \in \mathcal{N}\setminus\{a\}\ \} \end{aligned}$$

where $h(k,l,m) := \{(a,a), (\mathrm{E}_k(\mathrm{E}_l(m)), \mathrm{E}_k(m))\}$ and k,l,m are pairwise different wherever they occur, is a hedged bisimulation. $\mathcal{R}$ is "trivially" consistent as we never receive a nor the outermost keys of encrypted messages.

- As $P \xrightarrow{(\nu k,l,m)\,\overline{a}\langle \mathrm{E}_k(\mathrm{E}_l(m))\rangle.} (\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0})$ whenever k,l,m and a are pairwise different, we need to find $Q', N, \tilde{d}$ such that $Q \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} Q', a \notin \{\tilde{d}\}$ and $\mathcal{I}(\{(a,a), (\mathrm{E}_k(\mathrm{E}_l(m)), N)\}) \vdash (\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0})\ \mathcal{R}\ Q'$. We may choose $\tilde{d} = (k,m)$, $N = \mathrm{E}_k(m)$ and $Q' = \overline{a}\langle m\rangle.\mathbf{0}$. We also have that $\mathcal{I}(h(k,l,m)) = h(k,l,m)$.
- The output transitions of Q can be handled in the same way.
- As $(\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0}) \xrightarrow{a\,l} \mathbf{0}$ we must choose a matching transition of $\overline{a}\langle m\rangle.\mathbf{0}$. The only transition is valid.
- As $(\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0}) \xrightarrow{a\,m} \mathbf{0}$ we must choose a matching transition of $\overline{a}\langle m\rangle.\mathbf{0}$. The only transition is valid.
- As $\overline{a}\langle m\rangle.\mathbf{0} \xrightarrow{a\,m} \mathbf{0}$ we must choose a matching transition of the process $(\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0})$. Both transitions are valid.

The character of hedges is visible in the possible addition of the pair (l,m) in the last row of the above bisimulation relation; l and m may be different, but the hedge notes that they should correspond. □

Proposition 4.2.2 $(\{a\},\emptyset) \not\vdash P \approx_{\mathrm{f}} Q$.

Proof. By contradiction: assume that $(\{a\},\emptyset) \vdash P \approx_{\mathrm{f}} Q$.

- As a is in the frame and $P \xrightarrow{(\nu k,l,m)\,\overline{a}\langle \mathrm{E}_k(\mathrm{E}_l(m))\rangle.} (\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0})$ whenever k,l,m and a are pairwise different, we need to check that for all such $k,l,m \neq a$ there exist $Q', N, \tilde{d}, \mathrm{fr}, \mathrm{th}$ such that $Q \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} Q', a \notin \{\tilde{d}\}$, $a \in \mathrm{fr}$ and $(\mathrm{fr},\mathrm{th}) \vdash (\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0}) \approx_{\mathrm{f}} Q'$.
  Any transition of Q is of the form $Q \xRightarrow{(\nu k',m')\,\overline{a}\ \mathrm{E}_{k'}(m')} \overline{a}\langle m'\rangle.\mathbf{0}$.
- Since $(\overline{a}\langle m\rangle.\mathbf{0} + \overline{a}\langle l\rangle.\mathbf{0}) \xrightarrow{a\,l} \mathbf{0}$ we need to check that $\overline{a}\langle m'\rangle.\mathbf{0}$ can do a matching transition. As the only possibility is $\overline{a}\langle m'\rangle.\mathbf{0} \xrightarrow{a\,m'} \mathbf{0}$ there is a consistent frame-theory pair $(\mathrm{fr}_1, \mathrm{th}_1)$ such that $(\mathrm{fr}_1, \mathrm{th}_1) \vdash l \leftrightarrow m'$, which can only be the case if $l = m' \in \mathrm{fr}_1$.

- Since $(\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle l\rangle.\,\mathbf{0}) \xrightarrow{\overline{a}\,m} \mathbf{0}$ we need to check that $\overline{a}\langle m'\rangle.\,\mathbf{0}$ can do a matching transition. As the only possibility is $\overline{a}\langle m'\rangle.\,\mathbf{0} \xrightarrow{\overline{a}\,m'} \mathbf{0}$ there is a consistent frame-theory pair $\mathrm{fr}_2, \mathrm{th}_2$ such that $(\mathrm{fr}_2, \mathrm{th}_2) \vdash m \leftrightarrow m'$, which can only be the case if $m = m' \in \mathrm{fr}_2$.
- We thus have that $m = l$, which is a contradiction.

□

Note that P contains nondeterministic choice, a construct that is not present in the original spi calculus. However, this example works equally well if we replace the nondeterminism as expressed by choice either by an input and two mutually exclusive guards, as in

$$a(x).(\,[\,x = a\,]\,\overline{a}\langle m\rangle.\,\mathbf{0} \mid [\,x = b\,]\,\overline{a}\langle l\rangle.\,\mathbf{0}\,),$$

or with communication over a private channel, as in

$$(\nu c)\,(\,\overline{c}\langle a\rangle.\,\mathbf{0} \mid c(x).\overline{a}\langle m\rangle.\,\mathbf{0} \mid c(x).\overline{a}\langle l\rangle.\,\mathbf{0}\,).$$

The details are left as an exercise to the reader.

4.3 Framed vs. Hedged — Encryption Should Be Perfect

The following example exhibits a striking deficiency of framed bisimulation that is remedied by hedged bisimulation: a frame-theory pair can distinguish between the plaintext of an encrypted message (n in the example) and another random piece of data (m in the example).

$$\begin{array}{lllllll} P & := & (\nu k, n)\,\overline{a}\langle \mathrm{E}_k(n)\rangle.P' & \quad P' & := & (\nu m)\,\overline{a}\langle m\rangle.\ \mathbf{0} & \\ Q & := & (\nu k, n)\,\overline{a}\langle \mathrm{E}_k(n)\rangle.Q' & \quad Q' & := & \overline{a}\langle n\rangle.\ \mathbf{0} & \quad \text{where } n \neq a. \end{array}$$

We show that $\{(a,a)\} \vdash P \approx_{\mathrm{h}} Q$ and $(\{a\},\emptyset) \not\vdash P \approx_{\mathrm{f}} Q$. We first study the same relations for just the processes P' and Q'. (Note in passing that $\mathrm{fn}(P,Q) \subseteq \mathrm{n}(\{(a,a)\})$, although $\mathrm{fn}(Q') \not\subseteq \mathrm{n}(\{(a,a)\})$.)

Proposition 4.3.1 $\{(a,a)\} \vdash P' \approx_{\mathrm{h}} Q'$.

Proof. We show that the relation

$$\mathcal{R} := \{(\{(a,a)\}, P', Q')\} \cup \{(\{(a,a),(m,n)\}, \mathbf{0}, \mathbf{0}) \mid m \in \mathcal{N} \setminus \{a\}\}$$

is a hedged bisimulation. $\mathcal{R}$ is consistent, since for all hedges h such that $h \in \pi_1(\mathcal{R})$ conditions 1 and 3 are trivially satisfied and condition 2 follows from $n \neq a \neq m$. As neither P' nor Q' do input or internal computation we only have to check the conditions for output. As the output is on a pair of names known to the environment we need to check whether the environment accepts the input.

- As $P' \xrightarrow{(\nu m)\,\overline{a}\,m} \mathbf{0}$, we check that for all $m \neq a$ there are $Q'', N, \tilde{d}$ with $Q' \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} Q''$, $\{a,n\} \cap \{\tilde{d}\} = \emptyset$ and $\mathcal{I}(\{(a,a),(m,N)\}) \vdash \mathbf{0}\ \mathcal{R}\ Q''$. Clearly, $\tilde{d}$ is empty, $N = n$ and $Q'' = \mathbf{0}$. Immediately, we also have that $\mathcal{I}(\{(a,a),(m,n)\}) = \{(a,a),(m,n)\}$. Finally, $\{(a,a),(m,n)\} \vdash \mathbf{0}\ \mathcal{R}\ \mathbf{0}$.
- As $Q' \xrightarrow{\overline{a}\,n} \mathbf{0}$, we need to check that there are $P'', N, \tilde{d}$ with $P \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} P'', a \notin \{\tilde{d}\}$ and $\mathcal{I}(\{(a,a),(n,N)\}) \vdash P''\ \mathcal{R}\ \mathbf{0}$. We may choose $\tilde{d} = n$, $N = n$ and $P'' = \mathbf{0}$. Clearly, by inspection, $\mathcal{I}(\{(a,a),(n,n)\}) \vdash \mathbf{0}\ \mathcal{R}\ \mathbf{0}$.

□

The result can be strengthened as follows:

Proposition 4.3.2 *Whenever h is a consistent hedge such that $(a,a) \in h$, $n \notin \pi_2(h)$ and $\pi_2(h) \cap \{\mathrm{E}_n(M) \mid M \in \mathcal{M}\} = \emptyset$ we have that $h \vdash P' \approx_{\mathrm{h}} Q'$.*

Proof. The previous proof also holds for

$$\mathcal{R} := \{(h, P', Q')\} \cup \{\, (h \cup \{(m,n)\}, \mathbf{0}, \mathbf{0}) \mid m \in \mathcal{N} \setminus \mathrm{n}(\pi_1(h)) \,\}$$

since $\mathcal{R}$ is a consistent hedged relation by the preconditions. □

Note that in the above $\mathcal{R}$ the names m and n just correspond. They are previously unknown, can not be used as decryption keys, and there is no way to distinguish between them through further interaction with the process pair.

Since framed bisimilarity does not allow two different names to simply correspond, we have the following negative result.

Proposition 4.3.3 *There is no frame-theory pair* $(\mathrm{fr}, \mathrm{th})$ *such that* $a \in \mathrm{fr}$ *and* $(\mathrm{fr}, \mathrm{th}) \vdash P' \approx_{\mathrm{f}} Q'$.

Proof. Assume the opposite, and fix $m \neq n$ such that $m \in \mathcal{N} \setminus (\mathrm{fr} \cup \mathrm{n}(\pi_1(\mathrm{th})))$. As $a \in \mathrm{fr}$ and $P \xrightarrow{(\nu m)\,\overline{a}\,m} \mathbf{0}$ there must exist $Q'', N, \tilde{d}, \mathrm{fr}', \mathrm{th}'$ such that $Q' \overset{(\nu\tilde{d})\,\overline{a}\,N}{\Longrightarrow} Q''$, $(\mathrm{fr}', \mathrm{th}')$ is consistent, and $(\mathrm{fr}', \mathrm{th}') \vdash m \leftrightarrow N$. As the only transition of Q' is $Q' \xrightarrow{\overline{a}\,n} \mathbf{0}$ we have that $N = n$. Clearly SYN-ENC (cf. Def. 3.1.4) can not derive $(\mathrm{fr}', \mathrm{th}') \vdash m \leftrightarrow n$. Since $(\mathrm{fr}', \mathrm{th}')$ is consistent, which implies that $(m, n) \notin \mathrm{th}'$, we must have that $m = n \in \mathrm{fr}'$, which is a contradiction. □

Now, we can use the results for P' and Q' to derive results for P and Q.

Proposition 4.3.4 $\{(a, a)\} \vdash P \approx_{\mathrm{h}} Q$

Proof. As (a, a) is in the hedge, P and Q can do a matching output step. By Proposition 4.3.2, any resulting hedged process pair is in $\approx_{\mathrm{h}}$. □

Proposition 4.3.5 $(\{a\}, \emptyset) \not\vdash P \approx_{\mathrm{f}} Q$

Proof. As a is in the frame, P and Q can do a matching output step. By Proposition 4.3.3, no resulting framed process pair is in $\approx_{\mathrm{f}}$. □

An Up-To Interpretation.

According to [BDP02], alley bisimulation is sound up to restriction, meaning that names that are not present in the knowledge of the environment can be restricted in one or both of the processes. Applying this to the previous example, we get that

$$(\{{}^{a}\!/_{x}\}, \{{}^{a}\!/_{x}\}) \vdash \overline{a}\langle m\rangle.\,\mathbf{0} \approx_{\mathrm{a}} \overline{a}\langle m\rangle.\,\mathbf{0} \Longrightarrow (\{{}^{a}\!/_{x}\}, \{{}^{a}\!/_{x}\}) \vdash (\nu m)\,\overline{a}\langle m\rangle.\,\mathbf{0} \approx_{\mathrm{a}} \overline{a}\langle m\rangle.\,\mathbf{0}$$

By the above example this property does not hold for framed bisimulation. Thus we have:

Corollary 4.3.6 *Framed bisimulation is not sound up to restriction.*

5 Intermezzo

Having seen some examples that distinguish the bisimilarities of Section 3, we now try to establish as tight as possible formal relations between these bisimilarities.

5.1 Comparing (Standard) Bisimilarities

We recall some standard terminology used to compare bisimilarities in standard process calculi like the pi calculus. For instance, [MPW93] compared early and late bisimilarity (here denoted by $\sim_{\text{early}}$ and $\sim_{\text{late}}$) for the π-calculus by means of an *auxiliary* bisimilarity (here denoted by $\sim_{\text{aux}}$). Their findings can be summarized in various terminologies. Let P and Q be two processes.

1. Comparing early and auxiliary:
   - $P \sim_{\text{early}} Q$ iff $P \sim_{\text{aux}} Q$.
   - $\sim_{\text{early}} = \sim_{\text{aux}}$.
   - $\sim_{\text{early}}$ "is *equivalent* to" $\sim_{\text{aux}}$.
2. Comparing late and auxiliary:
   - $P \sim_{\text{aux}} Q$ if $P \sim_{\text{late}} Q$. (Likewise: $P \sim_{\text{late}} Q$ only if $P \sim_{\text{aux}} Q$.)
   - $\sim_{\text{late}} \subseteq \sim_{\text{aux}}$.
   - $\sim_{\text{late}}$ "is *sound* w.r.t." $\sim_{\text{aux}}$. (Likewise: $\sim_{\text{aux}}$ "is *complete* w.r.t." $\sim_{\text{late}}$.)

All these comparisons are simple, because the respective bisimilarities are sets of pairs of mere process terms. The situation gets less trivial when we consider environment-sensitive bisimilarities, which are sets of triples where the components play different roles.

5.2 Comparing Environment-Sensitive Bisimilarities

We first provide some meta-variables denoting bisimilarities and their corresponding environments. Let $\approx_{\text{x}}$ and $\approx_{\text{y}}$ denote any bisimilarities with corresponding environments e_{x} and e_{y}. We write $e_{\text{x}} \vdash P \approx_{\text{x}} Q$ for $(e_{\text{x}}, P, Q) \in \approx_{\text{x}}$ and $e_{\text{x}} \not\vdash P \approx_{\text{x}} Q$ otherwise.

As an example, let us recall the comparison of framed and fenced bisimilarities by [EHHO99]. Since these two bisimilarities have the same kind of environment, the comparison could simply be done by set-theoretic mutual inclusion checks, as in the case of standard bisimilarities.

This is no longer the case when comparing framed/fenced to their hedged and alley counterparts; we must find a more refined way to compare environment-sensitive bisimilarities. Every environment—independent of the particular kind of data structure involved—straightforwardly induces a binary relation on processes. E.g., to a frame-theory pair (fr, th) we may associate the set $\{(P,Q) \mid (\text{fr}, \text{th}) \vdash P \approx_{\text{f}} Q\}$. More generally, we may let $\mathcal{R}_{\text{x}}^{e_{\text{x}}}$ be the set $\{(P,Q) \mid e_{\text{x}} \vdash P \approx_{\text{x}} Q\}$. We can then say that an environment e_{y} is sound with respect to e_{x} if e_{y} does not relate any processes not related by e_{x}, i.e., if the set-theoretic inclusion $\mathcal{R}_{\text{y}}^{e_{\text{y}}} \subseteq \mathcal{R}_{\text{x}}^{e_{\text{x}}}$ holds.

It does not make much sense to compare $\mathcal{R}_{\text{y}}^{e_{\text{y}}}$ and $\mathcal{R}_{\text{x}}^{e_{\text{x}}}$ for unrelated e_{y} and e_{x}. However, if $\mathbf{E}_{\text{x}}$ and $\mathbf{E}_{\text{y}}$ are the sets of environments of the bisimilarities $\approx_{\text{x}}$ and $\approx_{\text{y}}$, then every function $g : \mathbf{E}_{\text{x}} \to \mathbf{E}_{\text{y}}$ gives us an embedding of the environments of $\approx_{\text{x}}$ into those of $\approx_{\text{y}}$. We may call such an embedding g sound (pointwise), if $\mathcal{R}_{\text{y}}^{g(e_{\text{x}})} \subseteq \mathcal{R}_{\text{x}}^{e_{\text{x}}}$ for all $e_{\text{x}} \in \mathbf{E}_{\text{x}}$; and we may call it complete (pointwise), if $\mathcal{R}_{\text{y}}^{g(e_{\text{x}})} \supseteq \mathcal{R}_{\text{x}}^{e_{\text{x}}}$ for all $e_{\text{x}} \in \mathbf{E}_{\text{x}}$. Along the same line, we may be inclined to call such an embedding g a (pointwise) bisimilarity equivalence if $\mathcal{R}_{\text{y}}^{g(e_{\text{x}})} = \mathcal{R}_{\text{x}}^{e_{\text{x}}}$ for all $e_{\text{x}} \in \mathbf{E}_{\text{x}}$. However, this definition does not yield symmetry: there may be environments in $\mathbf{E}_{\text{y}}$ that have no equivalent counterpart in $\mathbf{E}_{\text{x}}$. Instead, if $\mathcal{R}_{\text{y}}^{g(e_{\text{x}})} = \mathcal{R}_{\text{x}}^{e_{\text{x}}}$ for all $e_{\text{x}} \in \mathbf{E}_{\text{x}}$ we may call g a *full abstraction*, because it leaves unchanged the set of process pairs it relates, which we consider to be the essential property of an environment. Regarding full abstractions as giving rise to a preorder, we may define an appopriate notion of bisimilarity equivalence—reminiscient of the standard algebraic way—as the kernel of this pre-order. If $g : \mathbf{E}_{\text{x}} \to \mathbf{E}_{\text{y}}$ and $h : \mathbf{E}_{\text{y}} \to \mathbf{E}_{\text{x}}$ are full abstractions then we call (g, h) a bisimilarity equivalence relating $\approx_{\text{x}}$ and $\approx_{\text{y}}$.

To summarize, we have the following definitions.

Definition 5.2.1 *Assume that $\approx_{\text{x}}$ and $\approx_{\text{y}}$ are environment-sensitive bisimilarities, where $\mathbf{E}_{\text{x}}$ and $\mathbf{E}_{\text{y}}$ denote the sets of environments of $\approx_{\text{x}}$ and $\approx_{\text{y}}$, respectively. Then we define the following relations between $\approx_{\text{x}}$ and $\approx_{\text{y}}$.*

Soundness: $\approx_{\mathrm{y}}$ *is* g-sound *w.r.t.* $\approx_{\mathrm{x}}$
if $g : \mathbf{E}_{\mathrm{x}} \to \mathbf{E}_{\mathrm{y}}$ *is such that* $\forall e, P, Q :\ \ g(e) \vdash P \approx_{\mathrm{y}} Q$ *implies* $\ e \vdash P \approx_{\mathrm{x}} Q$.
Completeness: $\approx_{\mathrm{y}}$ *is* g-complete *w.r.t.* $\approx_{\mathrm{x}}$
if $g : \mathbf{E}_{\mathrm{x}} \to \mathbf{E}_{\mathrm{y}}$ *is such that* $\forall e, P, Q :\ \ e \vdash P \approx_{\mathrm{x}} Q$ *implies* $\ g(e) \vdash P \approx_{\mathrm{y}} Q$.
Full abstraction: $\approx_{\mathrm{y}}$ *is* fully g-abstract *w.r.t.* $\approx_{\mathrm{x}}$
if $g : \mathbf{E}_{\mathrm{x}} \to \mathbf{E}_{\mathrm{y}}$ *is such that* $\forall e, P, Q :\ \ e \vdash P \approx_{\mathrm{x}} Q$ *iff* $\ g(e) \vdash P \approx_{\mathrm{y}} Q$.
Equivalence: $\approx_{\mathrm{x}}$ *and* $\approx_{\mathrm{y}}$ *are* (g, h)-equivalent
if $\approx_{\mathrm{x}}$ *is fully* g*-abstract w.r.t.* $\approx_{\mathrm{y}}$ *and* $\approx_{\mathrm{y}}$ *is fully* h*-abstract w.r.t.* $\approx_{\mathrm{x}}$.

Proposition 5.2.2 *Soundness, completeness and full abstraction are reflexive and transitive, in the following sense:*

- $\approx_{\mathrm{x}}$ *is* $\mathrm{Id}_{\mathbf{E}_{\mathrm{x}}}$*-sound (complete, fully abstract) w.r.t. itself.*
- *If* $g : \mathbf{E}_{\mathrm{x}} \to \mathbf{E}_{\mathrm{y}}$ *and* $h : \mathbf{E}_{\mathrm{y}} \to \mathbf{E}_{\mathrm{z}}$ *are such that* $\approx_{\mathrm{y}}$ *is* g*-sound (complete, fully abstract) w.r.t.* $\approx_{\mathrm{x}}$ *and* $\approx_{\mathrm{z}}$ *is* h*-sound (complete, fully abstract) with respect to* $\approx_{\mathrm{y}}$*, then* $\approx_{\mathrm{z}}$ *is* $(h \circ g)$*-sound (complete, fully abstract) w.r.t.* $\approx_{\mathrm{x}}$.

Bisimilarity equivalence is reflexive, symmetric and transitive.

- $\approx_{\mathrm{x}}$ *is* $(\mathrm{Id}_{\mathbf{E}_{\mathrm{x}}}, \mathrm{Id}_{\mathbf{E}_{\mathrm{x}}})$*-equivalent to itself.*
- *If* $\approx_{\mathrm{x}}$ *and* $\approx_{\mathrm{y}}$ *are* (g, h)*-equivalent,*
  then $\approx_{\mathrm{y}}$ *and* $\approx_{\mathrm{x}}$ *are* (h, g)*-equivalent.*
- *If* $\approx_{\mathrm{x}}$ *and* $\approx_{\mathrm{y}}$ *are* (g, h)*-equivalent and* $\approx_{\mathrm{y}}$ *and* $\approx_{\mathrm{z}}$ *are* (g', h')*-equivalent,*
  then $\approx_{\mathrm{x}}$ *and* $\approx_{\mathrm{z}}$ *are* $(g \circ g', h' \circ h)$*-equivalent.*

Above, a bisimilarity equivalence is constructed from two a priori unrelated full abstractions g and h. Further constraints, such as requiring g and h to be inverse to each other, could be added to make the relation stronger. However, adding constraints on the functions causes problems for the robustness and transitivity of bisimilarity equivalence. It is not reasonable to require f and g to be inverses, since this can be broken by merely adding a "behavioral copy" of an environment to one of the bisimilarities. A weaker variant of this constraint is to require idempotence of $g \circ h$ and/or $h \circ g$, but it causes problems for transitivity — assuming that $g \circ h$ and $g' \circ h'$ are idempotent, it may well be that $g \circ g' \circ h' \circ h$ is not. (A counterexample is left as an exercise to the reader.)

5.3 Trivializing Soundness and Completeness

As it turns out, the definitions of g-soundness and g-completeness as defined in Definition 5.2.1 can be satisfied by trivial environment mappings. For completeness, the mapping between environments might collapse all environments of $\mathbf{E}_{\mathrm{x}}$ to a single trivial environment in $\mathbf{E}_{\mathrm{y}}$ that is equating any pair of processes, being "blind" for any possible distinction. For soundness we have the dual, namely that all environments of $\mathbf{E}_{\mathrm{x}}$ may be mapped to an environment in $\mathbf{E}_{\mathrm{y}}$ that discriminates between all process pairs.

Definition 5.3.1 *An environment* b *is* $\approx_{\mathrm{y}}$-blind *if* $\ b \vdash P \approx_{\mathrm{y}} Q$ *for all processes* P *and* Q.

Proposition 5.3.2 *If there is a* $\approx_{\mathrm{y}}$*-blind environment* b_{y} *and* $B_{\mathrm{y}} : \mathbf{E}_{\mathrm{x}} \to \mathbf{E}_{\mathrm{y}}$ *has* $\mathrm{range}(B_{\mathrm{y}}) = \{b_{\mathrm{y}}\}$ *then* $\approx_{\mathrm{y}}$ *is* B_{y}*-complete w.r.t.* $\approx_{\mathrm{x}}$.

Proof. Whenever $\ e_{\mathrm{x}} \vdash P \approx_{\mathrm{x}} Q$ we have $\ B_{\mathrm{y}}(e_{\mathrm{x}}) \vdash P \approx_{\mathrm{y}} Q$. □

Proposition 5.3.3 *All the previously defined bisimilarities have blind environments:*

1. $(\emptyset, \emptyset)$ *is a blind consistent frame-theory pair.*
2. $(\emptyset, \emptyset)$ *is a blind pair of equivalent substitutions.*
3. $\emptyset$ *is a blind consistent hedge.*

Proof. All bisimilarities are weak, so we only need to check that all detected process actions preserve the consistency of the environment. This is trivially true if no process actions can be detected by the environment.

1. A frame-theory pair with an empty frame can not detect any process actions. Obviously, $(\emptyset, \emptyset)$ is consistent.
2. There is no expression η such that $\mathrm{n}(\eta) \subseteq \mathrm{dom}(\emptyset) = \emptyset$. Trivially, $\emptyset \sim \emptyset$.
3. We have that $\mathcal{S}(\emptyset) = \emptyset$, so $\emptyset$ can not detect any process actions. Clearly, $\emptyset$ is a consistent hedge.

□

The above propositions imply the following completeness and full abstraction results.

Corollary 5.3.4 *Here we let B_{y} be some function that maps all environments to a single blind environment $\top$, as appropriate for its codomain. We also let $\approx_\top := \top \times \mathcal{P} \times \mathcal{P}$, i.e., where $\top \vdash P \approx_\top Q$ for all processes $P, Q \in \mathcal{P}$. Then*

1. *$\approx_{\mathrm{a}}$, $\approx_{\mathrm{f}}$, $\approx_{\#}$, $\approx_{\mathrm{h}}$ and $\approx_{\mathrm{s}}$ are B_{y}-complete w.r.t. $\approx_{\mathrm{a}}$, $\approx_{\mathrm{f}}$, $\approx_{\#}$, $\approx_{\mathrm{h}}$ and $\approx_{\mathrm{s}}$.*
2. *$\approx_{\mathrm{a}}$, $\approx_{\mathrm{f}}$, $\approx_{\#}$, $\approx_{\mathrm{h}}$ and $\approx_{\mathrm{s}}$ are fully B_{y}-abstract w.r.t. $\approx_\top$.*

The closest candidate to a dual of blindness turns out to be inconsistency. For the sake of exhibiting this duality, we assume that inconsistent environments are contained in the environment sets of the bisimilarities, although they don't relate any process pairs.

Proposition 5.3.5 *If there is a ($\approx_{\mathrm{y}}$-)inconsistent environment c_{y} and $C_{\mathrm{y}} : \mathbf{E}_{\mathrm{x}} \to \mathbf{E}_{\mathrm{y}}$ has* $\mathrm{range}(C_{\mathrm{y}}) = \{c_{\mathrm{y}}\}$ *then $\approx_{\mathrm{y}}$ is C_{y}-sound w.r.t. $\approx_{\mathrm{x}}$.*

Proof. Whenever $C_{\mathrm{y}}(e_{\mathrm{x}}) \vdash P \approx_{\mathrm{y}} Q$ (i.e., never), we have that $e_{\mathrm{x}} \vdash P \approx_{\mathrm{x}} Q$. □

As we have seen, there are inconsistent environments of all types, so if inconsistent environments are assumed to be in the environment domains we get this dual to Corollary 5.3.4.

Corollary 5.3.6 *Here we let C_{y} be some function that maps all environments to a single inconsistent environment $\bot$, as appropriate for its codomain. We also let $\approx_\bot := \bot \times \emptyset \times \emptyset$, i.e., where $\bot \not\vdash P \approx_\bot Q$ for all processes $P, Q \in \mathcal{P}$. Then*

1. *$\approx_{\mathrm{a}}$, $\approx_{\mathrm{f}}$, $\approx_{\#}$, $\approx_{\mathrm{h}}$ and $\approx_{\mathrm{s}}$ are C_{y}-sound w.r.t. $\approx_{\mathrm{a}}$, $\approx_{\mathrm{f}}$, $\approx_{\#}$, $\approx_{\mathrm{h}}$ and $\approx_{\mathrm{s}}$.*
2. *$\approx_{\mathrm{a}}$, $\approx_{\mathrm{f}}$, $\approx_{\#}$, $\approx_{\mathrm{h}}$ and $\approx_{\mathrm{s}}$ are fully C_{y}-abstract w.r.t. $\approx_\bot$.*

5.4 Good and Bad Environment Mappings

As we saw above, functions that collapse all environments to a single trivial environment give us the rather non-intuitive result that all non-trivial bisimilarities are sound and complete with respect to each other. To avoid this particular kind of trivialization, we want environment mappings that are sound for blindness (i.e., an environment is blind if its image is blind) and inconsistency.

Another property of all non-trivial environments is synthesis, i.e., the set of message pairs that the environment considers to be equal. As we can see in the definitions of the bisimulations, the synthesis of the environment is fundamental for its interactions with process pairs. On process input, the environment can only generate message pairs in its synthesis (possibly inventing some fresh names), and on process output, the environment cannot accept a message pair which matches one in the synthesis on only one side. We now proceed to prove that non-trivial environments relating the same processes must have the same synthesis.

For the rest of this section, we consider only non-trivial environments, i.e., frame-theory pairs, alleys and hedges and their corresponding bisimilarities of Section 3. To compare the synthesis of environments of different types, we extend Definition 3.1.5 as follows.

Definition 5.4.1 *If e_{x} and e_{y} are environments, than we write that $e_{\mathrm{x}} \leq e_{\mathrm{y}}$ if $e_{\mathrm{x}} \vdash M \leftrightarrow N$ implies that $e_{\mathrm{y}} \vdash M \leftrightarrow N$. We say that e_{x} and e_{y} are $\mathcal{M}$-*equivalent*, written $e_{\mathrm{x}} \gtreqless e_{\mathrm{y}}$, if $e_{\mathrm{x}} \leq e_{\mathrm{y}}$ and $e_{\mathrm{y}} \leq e_{\mathrm{x}}$.*

As we are mainly interested in full abstractions, it is convenient to have a shorthand for "e_{x} and e_{y} relate the same processes".

Definition 5.4.2 *Two environments e_{x} and e_{y} are $\approx^{\mathrm{x}}_{\mathrm{y}}$*-equivalent*, written $e_{\mathrm{x}} \equiv^{\mathrm{x}}_{\mathrm{y}} e_{\mathrm{y}}$ if for all processes P and Q we have that $e_{\mathrm{x}} \vdash P \approx_{\mathrm{x}} Q$ if and only if $e_{\mathrm{y}} \vdash P \approx_{\mathrm{y}} Q$.*

The relation between $\mathcal{M}$-equivalence and $\approx^{\mathrm{x}}_{\mathrm{y}}$-equivalence is given by the following proposition. This result implies that in order to be a full abstraction, the environment mapping must be faithful for blindness and consistency, and map a consistent non-blind environment to a $\mathcal{M}$-equivalent counterpart.

Proposition 5.4.3 *If e_{x} and e_{y} are $\approx^{\mathrm{x}}_{\mathrm{y}}$-equivalent then either $e_{\mathrm{x}} \gtrless e_{\mathrm{y}}$, e_{x} and e_{y} are both inconsistent, or e_{x} is $\approx_{\mathrm{x}}$-blind and e_{y} is $\approx_{\mathrm{y}}$-blind.*

Proof. If e_{x} is blind, but e_{y} is not, then there exist P, Q such that $e_{\mathrm{y}} \vdash P \approx_{\mathrm{y}} Q$ does not hold. As e_{x} is blind, $e_{\mathrm{x}} \vdash P \approx_{\mathrm{x}} Q$, so $e_{\mathrm{x}} \not\equiv^{\mathrm{x}}_{\mathrm{y}} e_{\mathrm{y}}$. The case where e_{y} is blind, but e_{x} is not, is handled in the same way.

If e_{x} is inconsistent, but e_{y} is not, then $e_{\mathrm{x}} \vdash \mathbf{0} \approx_{\mathrm{x}} \mathbf{0}$ does not hold. As e_{y} is consistent, $e_{\mathrm{y}} \vdash \mathbf{0} \approx_{\mathrm{y}} \mathbf{0}$, so $e_{\mathrm{x}} \not\equiv^{\mathrm{x}}_{\mathrm{y}} e_{\mathrm{y}}$. The case where e_{y} is inconsistent, but e_{x} is not, is handled in the same way.

If e_{x} is neither blind nor inconsistent then there exist names a, b such that $e_{\mathrm{x}} \vdash a \leftrightarrow b$. Let $P_1 = \overline{a}\langle a\rangle.\,\mathbf{0}$ and $Q_1 = (\nu k)\,\overline{b}\langle k\rangle.\,\mathbf{0}$. We have that $e_{\mathrm{x}} \nvdash P_1 \approx_{\mathrm{x}} Q_1$, since e_{x} can see the difference between a known name and a fresh name[2]. Since $e_{\mathrm{x}} \equiv^{\mathrm{x}}_{\mathrm{y}} e_{\mathrm{y}}$ we get that $e_{\mathrm{y}} \nvdash P_1 \approx_{\mathrm{y}} Q_1$, which implies that $e_{\mathrm{y}} \vdash a \leftrightarrow n$ or $e_{\mathrm{y}} \vdash n \leftrightarrow b$ for some name n. However, if we let $P_2 = a(z).\,\mathbf{0}$ and $Q_2 = b(z).\,\mathbf{0}$ we have that $e_{\mathrm{x}} \vdash P_2 \approx_{\mathrm{x}} Q_2$. Since $e_{\mathrm{x}} \equiv^{\mathrm{x}}_{\mathrm{y}} e_{\mathrm{y}}$ we get that $e_{\mathrm{y}} \vdash P_2 \approx_{\mathrm{y}} Q_2$, which implies that $e_{\mathrm{y}} \vdash a \leftrightarrow b$.

Now assume that $e_{\mathrm{x}} \vdash M \leftrightarrow N$. Let $P_3 = a(x).[\,x = M\,]\overline{a}\langle a\rangle.\,\mathbf{0}$ and $Q_3 = b(x).\,\mathbf{0}$. We have that $e_{\mathrm{x}} \nvdash P_3 \approx_{\mathrm{x}} Q_3$, since e_{x} can create and send M. Since $e_{\mathrm{x}} \equiv^{\mathrm{x}}_{\mathrm{y}} e_{\mathrm{y}}$ we get that $e_{\mathrm{y}} \nvdash P_3 \approx_{\mathrm{y}} Q_3$, which implies that there is e'_{y}, obtained from e_{y} by adding fresh names, such that $e'_{\mathrm{y}} \vdash M \leftrightarrow N'$ for some N'. As $\mathrm{n}(M) \subseteq \mathrm{fn}(P_3)$ we have that no name in $\mathrm{n}(M)$ can be created as fresh, so actually $e_{\mathrm{y}} \vdash M \leftrightarrow N'$.

Let $P_4 = \overline{a}\langle M\rangle.\,\mathbf{0}$ and $Q_4 = \overline{b}\langle N\rangle.\,\mathbf{0}$. Since $e_{\mathrm{x}} \vdash P_4 \approx_{\mathrm{x}} Q_4$ we must have $e_{\mathrm{y}} \vdash P_4 \approx_{\mathrm{y}} Q_4$ which is true only if $e_{\mathrm{y}} \vdash M \leftrightarrow N$, since a consistent environment can not consider M equivalent to two different messages N and N'.

We have now shown that $e_{\mathrm{x}} \leq e_{\mathrm{y}}$. By symmetry, we have that $e_{\mathrm{x}} \gtrless e_{\mathrm{y}}$ when $e_{\mathrm{x}} \equiv^{\mathrm{x}}_{\mathrm{y}} e_{\mathrm{y}}$ and neither e_{x} nor e_{y} is blind nor inconsistent. □

To summarize: An environment mapping should be sound for blindness and inconsistency and preserves the synthesis. We now turn to the task of finding such mappings, and verifying whether they are full abstractions.

6 Comparing Environments

Our relations on environment-sensitive bisimulations are based on a comparison of the various environments. In this section, we introduce mappings between frame-theory pairs (**FT**), hedges (**H**) and substitution pairs (**SS**).

$$\begin{tikzcd}
\mathbf{FT} \arrow[d, "\psi"] & \\
\mathbf{H} \arrow[r, shift left, "\theta^f"] & \mathbf{SS} \arrow[l, shift left, "\varphi"] \arrow[ul, "\gamma"']
\end{tikzcd}$$

[2] Compare with Proposition 4.3.3 and Proposition 4.3.2, noting the difference between a *known* and a *used* name.

Accompanying these mappings, we assemble a "toolbox" of auxiliary results that are used to relate hedges and the other kinds of environments, notably frames and fences.

6.1 Properties of Hedges

As we intend to use hedged bisimilarity, which we have introduced in this paper, as the yardstick against which all other bisimularities are measured, a preliminary investigation of the properties of hedges is offered, and indeed exploited in the proofs later on.

From the definition of hedges, we immediately get

Lemma 6.1.1 *If* $h \vdash M \leftrightarrow N$ *and* $M \in \mathcal{N}$ *or* $N \in \mathcal{N}$*, then* $(M, N) \in h$.

Proof. Clearly Syn-enc can not derive $h \vdash M \leftrightarrow N$. □

We define a pre-ordering on hedges as follows:

Definition 6.1.2 $g \leq h$ *iff* $\mathcal{S}(g) \subseteq \mathcal{S}(h)$*. If* $g \leq h$ *and* $h \leq g$ *we say that* g *and* h *are* $\mathcal{M}$-equivalent*, written* $g \gtreqless h$*.*

$\leq$ is transitive and reflexive by the same properties for $\subseteq$. For results on antisymmetry, see Corollary 6.1.12 and this example:

Example 6.1.3 *Let*

$$\begin{aligned} g_1 &= \{(a,a),(c,k),(\mathrm{E}_k(b),\mathrm{E}_l(a)),(\mathrm{E}_a(c),\mathrm{E}_a(k))\} \\ g_2 &= \{(a,a),(c,k),(\mathrm{E}_k(b),\mathrm{E}_l(a)),(\mathrm{E}_c(\mathrm{E}_k(b)),\mathrm{E}_k(\mathrm{E}_l(a)))\} \end{aligned}$$

Then $g_1 \gtreqless g_2$*. More generally, if* h *is a hedge,* $h_1 \leq h$ *and* $h_2 \leq h$ *then* $h \cup h_1 \gtreqless h \cup h_2$*.*

An alternative definition of $\leq$ is as follows:

Lemma 6.1.4 $g \leq h$ *iff* $g \subseteq \mathcal{S}(h)$

Proof. $g \leq h \Rightarrow g \subseteq \mathcal{S}(h)$, since $g \subseteq \mathcal{S}(g) \subseteq \mathcal{S}(h)$.

For the other direction, assume that $g \subseteq \mathcal{S}(h)$ and take any M, N such that $g \vdash M \leftrightarrow N$. By induction on the derivation of $g \vdash M \leftrightarrow N$, we have that $h \vdash M \leftrightarrow N$. □

Corollary 6.1.5 *Some properties relating* $\leq$ *and set operations:*

1. *If* $g \subseteq h$ *then* $g \leq h$*.*
2. *If* $g \leq h$ *and* $f \leq h$ *then* $(f \cup g) \leq h$*.*
3. *If* $g_1 \leq h_1$ *and* $g_2 \leq h_2$ *then* $(g_1 \cup g_2) \leq (h_1 \cup h_2)$*.*

Irreducible Hedges

An interesting subset of **H** is the set of *irreducible* hedges. These are intuitively hedges that are reduced as far as possible, in the sense that no pair of messages in them can be decrypted using the information in the hedge. These results are later used to relate hedges with other types of environments, especially regarding how the environment reacts to process output.

Definition 6.1.6 *A hedge* h *is* irreducible *if* $h = \mathcal{I}(h)$*.* h *is* reducible *if* h *is not irreducible.*

An alternative definition is as follows:

Lemma 6.1.7 *A hedge* h *is irreducible iff the following condition holds:*
If $(\mathrm{E}_a(M), \mathrm{E}_b(N)) \in h$ *then* $(a, b) \notin h$*.*

Proof. If this holds then we can not apply Hedge-dec to any pair in h, so $\mathcal{A}(h) = h$. By the definition of $\mathcal{I}(h)$ we then have that $\mathcal{I}(h) = h$.

If h is irreducible then the condition holds by the definition of $\mathcal{I}(h)$. □

Corollary 6.1.8 *$\mathcal{I}(h)$ is irreducible for all hedges h.*

Example 6.1.9 *All hedges defined in Example 3.3.3 are irreducible, but neither of the hedges defined in Example 6.1.3. Moreover, if g is reducible and h is any hedge, then $g \cup h$ is reducible.*

As might be expected, the irreducibles of a hedge can be used to generate any message that can be generated by the hedge.

Lemma 6.1.10 *For any hedge h, $h \leq \mathcal{A}(h) \gtreqless \mathcal{I}(h)$.*

Proof. As $h \subseteq \mathcal{A}(h) \supseteq \mathcal{I}(h)$ Corollary 6.1.5(1) gives that $h \leq \mathcal{A}(h) \geq \mathcal{I}(h)$. What remains to be proved is $\mathcal{A}(h) \leq \mathcal{I}(h)$. By Lemma 6.1.4, it suffices to show $\mathcal{A}(h) \subseteq \mathcal{S}(\mathcal{I}(h))$. Assuming that $(M, N) \in \mathcal{A}(h)$, we can show that $\mathcal{I}(h) \vdash M \leftrightarrow N$ by structural induction on M. □

An irreducible hedge is a subset of any $\mathcal{M}$-equivalent hedge.

Lemma 6.1.11 *If $g \gtreqless h$ and g is irreducible then $g \subseteq h$.*

Proof. Take any $(M, N) \in g$. As $g \gtreqless h$, $h \vdash M \leftrightarrow N$. We have two cases:

If $M \in \mathcal{N}$ or $N \in \mathcal{N}$ then $(M, N) \in h$ by Lemma 6.1.1.

Else, $M = \mathrm{E}_a(M')$ and $N = \mathrm{E}_b(N')$. Since g is irreducible $(a, b) \notin g$ by Lemma 6.1.7. By Lemma 6.1.1 $g \not\vdash a \leftrightarrow b$, so $h \not\vdash a \leftrightarrow b$ and Syn-enc can not derive $h \vdash M \leftrightarrow N$. This shows that $(M, N) \in h$. □

Two $\mathcal{M}$-equivalent irreducible hedges are equal, so the preorder $\leq$ is an ordering relation on the set of irreducible hedges.

Corollary 6.1.12 *If $g \gtreqless h$ and both g and h are irreducible, then $g = h$.*

The ordering of hedges is preserved by $\mathcal{I}(\cdot)$.

Lemma 6.1.13 *If $g \leq h$, then $\mathcal{I}(g) \leq \mathcal{I}(h)$.*

Proof. According to Lemma 6.1.10 $\mathcal{I}(g) \leq \mathcal{A}(g)$, so by the transitivity of $\leq$ we only need to show $\mathcal{A}(g) \leq \mathcal{I}(h)$. By Lemma 6.1.4 this holds iff $\mathcal{A}(g) \subseteq \mathcal{S}(\mathcal{I}(h))$, which we show by induction on the derivation of $\mathcal{A}(g)$. Take any $(M, N) \in \mathcal{A}(g)$.

The base case is that $(M, N) \in g$. By Lemma 6.1.10 $h \leq \mathcal{I}(h)$, so $g \leq \mathcal{I}(h)$ by the transitivity of $\leq$. In particular, $\mathcal{I}(h) \vdash M \leftrightarrow N$.

Otherwise we have used Ana-dec to derive $(M, N) \in \mathcal{A}(g)$, which means that there are a and b such that $(\mathrm{E}_a(M), \mathrm{E}_b(N)) \in \mathcal{A}(g)$ and $(a, b) \in \mathcal{A}(g)$. By induction $\mathcal{I}(h) \vdash \mathrm{E}_a(M) \leftrightarrow \mathrm{E}_b(N)$ and $\mathcal{I}(h) \vdash a \leftrightarrow b$. By Lemma 6.1.1 $(a, b) \in \mathcal{I}(h)$, so $(\mathrm{E}_a(M), \mathrm{E}_b(N)) \notin \mathcal{I}(h)$ by the definition of $\mathcal{I}(\cdot)$. Then Syn-enc must have been used to derive $\mathcal{I}(h) \vdash \mathrm{E}_a(M) \leftrightarrow \mathrm{E}_b(N)$, which gives that $\mathcal{I}(h) \vdash M \leftrightarrow N$. □

The irreducibles of two $\mathcal{M}$-equivalent hedges are equal.

Lemma 6.1.14 *If $g \gtreqless h$, then $\mathcal{I}(g) = \mathcal{I}(h)$.*

Proof. $\mathcal{I}(g) \gtreqless \mathcal{I}(h)$ by Lemma 6.1.13. $\mathcal{I}(g)$ and $\mathcal{I}(h)$ are both irreducible by Corollary 6.1.8. The equality then follows from Corollary 6.1.12. □

We also have this more general variant of Corollary 6.1.8, that is used to prove that hedges and alleys behave similarly on process output.

Lemma 6.1.15 *If g and h are hedges, then $\mathcal{I}(\mathcal{I}(h) \cup g) = \mathcal{I}(h \cup g)$.*

Proof. By Corollary 6.1.5(1) $g \leq h \cup g$, so using Lemma 6.1.13 we have that $\mathcal{I}(g) \leq \mathcal{I}(h \cup g)$. By Lemma 6.1.10 $g \leq \mathcal{I}(g)$, so by transitivity $g \leq \mathcal{I}(h \cup g)$. For $\mathcal{I}(h)$ we know that $\mathcal{I}(h) \subseteq \mathcal{A}(h) \subseteq \mathcal{A}(h \cup g) \leq \mathcal{I}(h \cup g)$, where the last relation is due to Lemma 6.1.10. By Corollary 6.1.5(2) $\mathcal{I}(h) \cup g \leq \mathcal{I}(h \cup g)$, so $\mathcal{I}(\mathcal{I}(h) \cup g) \leq \mathcal{I}(\mathcal{I}(h \cup g))$ by Lemma 6.1.13. $\mathcal{I}(h \cup g)$ is irreducible by Corollary 6.1.8, so $\mathcal{I}(\mathcal{I}(h) \cup g) \leq \mathcal{I}(h \cup g)$.

$h \leq \mathcal{I}(h)$ by Lemma 6.1.10, so by Corollary 6.1.5(3) we have that $h \cup g \leq \mathcal{I}(h) \cup g$. By Lemma 6.1.13 we have that $\mathcal{I}(h \cup g) \leq \mathcal{I}(\mathcal{I}(h) \cup g)$, so $\mathcal{I}(h \cup g) \gtreqless \mathcal{I}(\mathcal{I}(h) \cup g)$. The equality now follows from Corollary 6.1.12. □

Proper Care for Consistent Hedges

As we have seen in Section 5, an environment mapping should preserve consistency. In order to show that this holds for the environment mappings we will define, we investigate consistent hedges and their properties.

Lemma 6.1.16 *If h is consistent then h is irreducible.*

Proof. By condition 3 for consistency, we have that $(\mathrm{E}_a(M), \mathrm{E}_b(N)) \in h$ implies $(a, b) \notin h$. By Lemma 6.1.7 this means that h is irreducible. □

Note that we have only used a special case of condition 3 in the proof of Lemma 6.1.16. The three conditions for consistency are pairwise disjoint (see Example 3.3.3), so consistency is a much stronger constraint then irreducibility.

A generalized version of condition 2 for consistency is that a consistent hedge can not generate two message pairs that only differ in one component.

Lemma 6.1.17 *Let h be consistent with $h \vdash M \leftrightarrow N$ and $h \vdash M' \leftrightarrow N'$. Then $M = M'$ iff $N = N'$.*

Proof. By symmetry we need only study the case $M = M'$. The proof is by induction on the derivation of $h \vdash M \leftrightarrow N$.

If $(M, N) \in h$ we will first show that $(M, N') \in h$. If M is a name this follows from Lemma 6.1.1. Otherwise $M = \mathrm{E}_a(K)$, but since $a \notin \pi_1(h)$ by condition 3 of consistency we can not use Syn-enc to derive $h \vdash M \leftrightarrow N'$. Now we know that $(M, N) \in h$ and $(M, N') \in h$, so $N = N'$ by condition 2 for consistency.

If $M = \mathrm{E}_a(K)$, $N = \mathrm{E}_b(L)$, $h \vdash K \leftrightarrow L$ and $h \vdash a \leftrightarrow b$ then $a \in \pi_1(h)$ by Lemma 6.1.1. As h is consistent, $M \notin \pi_1(h)$, so we must have used Syn-enc to derive $h \vdash M \leftrightarrow N'$. This gives that $N' = \mathrm{E}_c(L')$ for some c, L' such that $h \vdash K \leftrightarrow L'$ and $h \vdash a \leftrightarrow c$. By induction $L = L'$ and $b = c$. □

Two $\mathcal{M}$-equivalent consistent hedges are always equal.

Lemma 6.1.18 *If $g \gtreqless h$ and both g and h are consistent, then $g = h$.*

Proof. g and h are irreducible by Lemma 6.1.16. The equality follows from Corollary 6.1.12. □

Any irreducible "trimming" of a consistent hedge is consistent.

Lemma 6.1.19 *If h is consistent, g is irreducible and $g \leq h$ then g is consistent.*

Proof. Assume that $(M, N) \in g$ and note that $h \vdash M \leftrightarrow N$. We only need to show one direction of the symmetric conditions.

1. If $M \in \mathcal{N}$ then $(M, N) \in h$ by Lemma 6.1.1, so $N \in \mathcal{N}$ as h is consistent.
2. See Lemma 6.1.17.

3. Assume that $M = \mathrm{E}_a(K)$. If $(M, N) \in h$ then $a \notin \pi_1(h)$ by condition 3 for consistency, so $a \notin \pi_1(g)$ by Lemma 6.1.1.
   Else Syn-enc has been used to derive $h \vdash M \leftrightarrow N$, so $N = \mathrm{E}_b(L)$ where $h \vdash a \leftrightarrow b$ and $h \vdash K \leftrightarrow L$. By Lemma 6.1.7 $(a, b) \in g$ would contradict that g is irreducible. If $N \neq b$ we have by Lemma 6.1.17 that $h \not\vdash a \leftrightarrow N$. This gives that $(a, N) \notin g$ so $a \notin \pi_1(g)$.

□

Disjoint consistent hedges may be directly combined.

Lemma 6.1.20 *If g and h are consistent and $\mathrm{n}(g) \cap \mathrm{n}(h) = \emptyset$, then $g \cup h$ is consistent.*

Proof. Take any message pair such that $(M, N) \in g \cup h$. By symmetry we may assume that $(M, N) \in g$.

1. $M \in \mathcal{N} \iff N \in \mathcal{N}$ is clear, since g is consistent.
2. Take any $(M', N') \in g \cup h$. As $\mathrm{n}(M) \neq \emptyset \neq \mathrm{n}(N)$ we have that $(M', N') \in g$ whenever $M = M'$ or $N = N'$. As g is consistent, $M = M'$ iff $N = N'$.
3. If $M = \mathrm{E}_a(M')$ and $N = \mathrm{E}_b(N')$ then $a \notin \pi_1(g)$ and $b \notin \pi_2(g)$ as g is consistent. As $\{a, b\} \subseteq \mathrm{n}(g)$ we have that $\{a, b\} \cap \mathrm{n}(h) = \emptyset$ and as a special case of this, $a \notin \pi_1(h)$ and $b \notin \pi_2(h)$.

□

6.2 Frames and Hedges

Since the definitions of hedges and frame-theory pairs are similar, the correspondence is fairly clear. However, as pointed out in Section 4, hedged and framed bisimilarity do not coincide. In Section 7, we will prove that framed bisimulation implies hedged bisimulation. We begin with some results relating frame-theory pairs and hedges.

Definition 6.2.1 *The hedge corresponding to a frame-theory pair is defined by:*

$$\psi : \mathbf{FT} \to \mathbf{H} : \psi(\mathrm{fr}, \mathrm{th}) := \mathrm{Id}_{\mathrm{fr}} \cup \mathrm{th}$$

Note that $\mathcal{S}(\mathrm{fr}, \mathrm{th}) = \mathcal{S}(\psi(\mathrm{fr}, \mathrm{th}))$, so $(\mathrm{fr}, \mathrm{th}) \leq (\mathrm{fr}', \mathrm{th}')$ iff $\psi(\mathrm{fr}, \mathrm{th}) \leq \psi(\mathrm{fr}', \mathrm{th}')$. We thus have that ψ preserves the synthesis of frame-theory pairs, which was one of the desireable properties mentioned in Section 5.

Since consistent frame-theory pairs have names only in the frame, we have this obvious strengthening of Lemma 6.1.1.

Lemma 6.2.2 *If $(\mathrm{fr}, \mathrm{th})$ is consistent and $(\mathrm{fr}, \mathrm{th}) \vdash M \leftrightarrow N$ where $M \in \mathcal{N}$ or $N \in \mathcal{N}$ then $M = N \in \mathrm{fr}$.*

Proof. Since Syn-enc can not derive $(\mathrm{fr}, \mathrm{th}) \vdash M \leftrightarrow N$ we have that $(M, N) \in \psi(\mathrm{fr}, \mathrm{th})$. There are no names in the theory by condition 1 for the consistency of $(\mathrm{fr}, \mathrm{th})$, so $M = N \in \mathrm{fr}$. □

An important result is that if a frame-theory pair is consistent then its corresponding hedge is also consistent. In other words, ψ preserves consistency.

Lemma 6.2.3 *If $(\mathrm{fr}, \mathrm{th})$ is consistent then $\psi(\mathrm{fr}, \mathrm{th})$ is consistent.*

Proof. Take any $(M, N) \in \psi(\mathrm{fr}, \mathrm{th})$. We show only one direction of the symmetric conditions.

1. $M \in \mathcal{N} \iff N \in \mathcal{N}$ by Lemma 6.2.2.
2. Assume that $(M, N') \in \psi(\mathrm{fr}, \mathrm{th})$. If $\{M, N, N'\} \cap \mathcal{N} \neq \emptyset$ then $M = N = N' \in \mathrm{fr}$ by Lemma 6.2.2. Else, $(M, N) \in \mathrm{th}$ and $(M, N') \in \mathrm{th}$, so $N = N'$ by condition 2 of the consistency of $(\mathrm{fr}, \mathrm{th})$.

3. If $M = \mathrm{E}_a(M')$ then according to Lemma 6.2.2 $a \in \pi_1(\psi(\mathrm{fr}, \mathrm{th}))$ only if $a \in \mathrm{fr}$, which is false by condition 3 for the consistency of $(\mathrm{fr}, \mathrm{th})$.

□

Note that the reverse implication does not hold. For example, $(\emptyset, \{(a, b)\})$ is not a consistent frame-theory pair, but $\psi(\emptyset, \{(a, b)\}) = \{(a, b)\}$ is a consistent hedge.

We will need an extension of Lemma 6.1.19 to consistent frame-theory pairs.

Lemma 6.2.4 *If* $(\mathrm{fr}, \mathrm{th})$ *is consistent,* h *is irreducible and* $h \leq \psi(\mathrm{fr}, \mathrm{th})$ *there exist* $\mathrm{fr}', \mathrm{th}'$ *such that* $h = \psi(\mathrm{fr}', \mathrm{th}')$ *and* $(\mathrm{fr}', \mathrm{th}')$ *is consistent.*

Proof. We show that $\mathrm{fr}' = \pi_1(h \cap (\mathcal{N} \times \mathcal{N})), \mathrm{th}' = h \setminus \mathrm{Id}_{\mathrm{fr}'}$ have the desired properties. Note that $h \subseteq \mathcal{S}(\psi(\mathrm{fr}, \mathrm{th}))$ by Lemma 6.1.4.

1. We first show that $\mathrm{Id}_{\mathrm{fr}'} = h \cap (\mathcal{N} \times \mathcal{M}) = h \cap (\mathcal{M} \times \mathcal{N})$. Take $(M, N) \in h$ such that $M \in \mathcal{N}$ or $N \in \mathcal{N}$. By Lemma 6.2.2 applied to $\psi(\mathrm{fr}, \mathrm{th}) \vdash M \leftrightarrow N$ we have that $M = N \in \mathrm{fr}$, so $M = N \in \mathrm{fr}'$ by definition. On the other hand, whenever $a \in \mathrm{fr}'$ then there is by definition b such that $(a, b) \in h$. As above $a = b$ by Lemma 6.2.2.
2. We then show that $\psi(\mathrm{fr}', \mathrm{th}') = h$. By the definition of $(\mathrm{fr}', \mathrm{th}')$ we have that $\psi(\mathrm{fr}', \mathrm{th}') = \mathrm{th}' \cup \mathrm{Id}_{\mathrm{fr}'} = (h \setminus \mathrm{Id}_{\mathrm{fr}'}) \cup \mathrm{Id}_{\mathrm{fr}'} = h \cup \mathrm{Id}_{\mathrm{fr}'} \supseteq h$. We have equality if $\mathrm{Id}_{\mathrm{fr}'} \subseteq h$, which follows from step 1.
3. Now we show the consistency of $(\mathrm{fr}', \mathrm{th}')$. First $\psi(\mathrm{fr}, \mathrm{th})$ is consistent by Lemma 6.2.3, so h is consistent by Lemma 6.1.19. Take any $(M, N) \in \mathrm{th}'$ and note that $\psi(\mathrm{fr}, \mathrm{th}) \vdash M \leftrightarrow N$.
   (a) If $M \in \mathcal{N}$ or $N \in \mathcal{N}$ then by step 1 we have that $(M, N) \notin h \setminus \mathrm{Id}_{\mathrm{fr}'} = \mathrm{th}'$.
   (b) If $(M', N') \in \mathrm{th}'$ then we have that $\psi(\mathrm{fr}, \mathrm{th}) \vdash M' \leftrightarrow N'$, so $M = M' \iff N = N'$ by Lemma 6.1.17.
   (c) Assume that $M = \mathrm{E}_a(M')$ and $N = \mathrm{E}_b(N')$. By the consistency of h we have that $a \notin \pi_1(h)$ and $b \notin \pi_2(h)$, so by step 1 we have that $a, b \notin \mathrm{fr}'$.

□

6.3 Fences and Hedges

Informally, ξ (see Table 4 on page 9) is a function that extends a given consistent frame-theory pair with a new pair of messages. If the resulting frame-theory pair is not consistent, ξ returns $\perp$. Since the results of [EHHO99] hold for a superset of $\mathcal{M}$ it is easy to see that they hold for the message syntax of this paper. To show the soundness of fenced bisimulation with respect to hedged we exhibit that ξ is a sound approximation of $\mathcal{I}(\cdot)$. We start with two results proved by [EHHO99].

Soundness: ξ always creates a consistent extension.

Lemma 6.3.1 *If* $(\mathrm{fr}, \mathrm{th})$ *is consistent and* $\xi(\mathrm{fr}, \mathrm{th}, M, N) \neq \perp$ *then*

1. $\xi(\mathrm{fr}, \mathrm{th}, M, N)$ *is consistent*
2. $\xi(\mathrm{fr}, \mathrm{th}, M, N) \vdash M \leftrightarrow N$
3. $(\mathrm{fr}, \mathrm{th}) \leq \xi(\mathrm{fr}, \mathrm{th}, M, N)$

Completeness: If there exists a consistent extension then ξ is defined and returns the smallest of all consistent extensions.

Lemma 6.3.2 *If* $(\mathrm{fr}, \mathrm{th})$ *and* $(\mathrm{fr}', \mathrm{th}')$ *are consistent,* $(\mathrm{fr}, \mathrm{th}) \leq (\mathrm{fr}', \mathrm{th}')$ *and* $(\mathrm{fr}', \mathrm{th}') \vdash M \leftrightarrow N$ *then* $\xi(\mathrm{fr}, \mathrm{th}, M, N) \neq \perp$ *and* $\xi(\mathrm{fr}, \mathrm{th}, M, N) \leq (\mathrm{fr}', \mathrm{th}')$.

Now we can show the relationship between ξ and $\mathcal{I}(\cdot)$, namely that whenever a consistent fence accepts a message pair, its corresponding hedge also does. That the converse does not hold was shown in Section 4.

Lemma 6.3.3 *If* $(\mathrm{fr}, \mathrm{th})$ *is consistent and* $\xi(\mathrm{fr}, \mathrm{th}, M, N) \neq \perp$ *then*

$$\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}) = \psi(\xi(\mathrm{fr}, \mathrm{th}, M, N))$$

Proof. Let $g = \psi(\mathrm{fr}, \mathrm{th})$, $g' = g \cup \{(M, N)\}$ and $h_\xi = \psi(\xi(\mathrm{fr}, \mathrm{th}, M, N))$. We first show that $\mathcal{I}(g') \leq h_\xi$, then the other direction.

By Lemma 6.3.1 $\xi(\mathrm{fr}, \mathrm{th}, M, N) \vdash M \leftrightarrow N$ and $(\mathrm{fr}, \mathrm{th}) \leq \xi(\mathrm{fr}, \mathrm{th}, M, N)$. After application of ψ this gives that $h_\xi \vdash M \leftrightarrow N$ and $g \leq h_\xi$, so the combined hedge $g' \leq h_\xi$ by Corollary 6.1.5(3). Reducing both sides, Lemma 6.1.13 gives $\mathcal{I}(g') \leq \mathcal{I}(h_\xi)$. By Lemma 6.3.1 we have that $\xi(\mathrm{fr}, \mathrm{th}, M, N)$ is consistent, so h_ξ is consistent by Lemma 6.2.3. h_ξ is then irreducible by Lemma 6.1.16, so $\mathcal{I}(g') \leq h_\xi$.

Since $\mathcal{I}(g')$ is irreducible by Corollary 6.1.8 we can apply Lemma 6.2.4 to $\mathcal{I}(g') \leq h_\xi$, giving us a consistent $(\mathrm{fr}', \mathrm{th}')$ such that $\mathcal{I}(g') = \psi(\mathrm{fr}', \mathrm{th}')$. We have by Lemma 6.1.10 that $g' \leq \mathcal{I}(g')$, which can be divided into $\mathcal{I}(g') \vdash M \leftrightarrow N$ and $g \leq \mathcal{I}(g')$. In the framed world, this means that $(\mathrm{fr}', \mathrm{th}') \vdash M \leftrightarrow N$ and $(\mathrm{fr}, \mathrm{th}) \leq (\mathrm{fr}', \mathrm{th}')$. Then we can apply Lemma 6.3.2, which gives that $\xi(\mathrm{fr}, \mathrm{th}, M, N) \leq (\mathrm{fr}', \mathrm{th}')$. Going back to the hedges, this means that $h_\xi \leq \mathcal{I}(g')$.

We have now proved the inequality in both directions, so $h_\xi \gtreqless \mathcal{I}(g')$ and the equality follows from Corollary 6.1.12. □

6.4 Hedges vs. Alleys

In this section we show that hedges are in a certain sense equivalent to the environments of alley bisimulation. As the theories are fairly different, we need a number of auxiliary results to tie the two representations together.

Lemmas on Substitutions

We first recall some lemmas proved by [BDP02]:

Lemma 6.4.1 $\forall \sigma, x \in \mathrm{dom}(\sigma) : \exists \zeta : \mathrm{n}(\zeta) \subseteq \mathrm{dom}(\sigma) \wedge [\![\zeta\sigma]\!] = \mathcal{C}(\sigma, x)$.

Lemma 6.4.2 *For* $\sigma \sim \rho$ *where* $\mathrm{dom}(\sigma) = \{x_i\}_{i \in I}$ *we let* $M_i = \mathcal{C}(\sigma, x_i)$ *and* $N_i = \mathcal{C}(\rho, x_i)$. *Then, for every* ζ *such that* $\mathrm{n}(\zeta) \subseteq \mathrm{dom}(\sigma)$ *either:*

1. $[\![\zeta\sigma]\!] = [\![\zeta\rho]\!] = \perp$, *or*
2. *There are* i *and a tuple* $\tilde{\iota}$ *of indices from* I *such that*

$$\begin{array}{rcl} [\![\zeta\sigma]\!] & = & \mathrm{E}_{M_{\iota_m}}(\cdots \mathrm{E}_{M_{\iota_2}}(\mathrm{E}_{M_{\iota_1}}(M_i)) \cdots) \\ [\![\zeta\rho]\!] & = & \mathrm{E}_{N_{\iota_m}}(\cdots \mathrm{E}_{N_{\iota_2}}(\mathrm{E}_{N_{\iota_1}}(N_i)) \cdots) . \end{array}$$

As an abbreviation, whenever $\mathrm{dom}(\sigma) = \{x_i\}_{i \in I}$ and $\{\tilde{\iota}\} \subseteq I$ we write $\mathrm{E}_{\tilde{\iota}}(\sigma_i)$ for $\mathrm{E}_{\mathcal{C}(\sigma, x_{\iota_m})}(\cdots \mathrm{E}_{\mathcal{C}(\sigma, x_{\iota_2})}(\mathrm{E}_{\mathcal{C}(\sigma, x_{\iota_1})}(\mathcal{C}(\sigma, x_i))) \cdots)$

Lemma 6.4.3 *If* $\sigma \sim \rho$ *and* $\mathrm{n}(\zeta) \subseteq \mathrm{dom}(\sigma)$ *with* $[\![\zeta\sigma]\!] \neq \perp$
then $\sigma\{{}^{[\![\zeta\sigma]\!]}/_{x}\} \sim \rho\{{}^{[\![\zeta\rho]\!]}/_{x}\}$.

Lemma 6.4.4 *If* $\sigma\{{}^{M}/_{x}\} \sim \rho\{{}^{N}/_{x}\}$ *then* $\sigma \sim \rho$.

Lemma 6.4.5 *If* $M \in \mathcal{A}(\sigma)$ *then there are* i *and* $\tilde{\iota}$ *such that* $M = \mathrm{E}_{\tilde{\iota}}(\sigma_i)$.

Lemma 6.4.6 $\mathrm{fn}(\sigma) = \mathrm{n}(\mathcal{I}(\sigma))$

From Alleys to Hedges

We first need some natural way to move back and forth between consistent hedges and consistent pairs of substitutions.

Given a consistent pair of substitutions, we can use the cores to get a hedge as follows:

Definition 6.4.7 *The hedge corresponding to a substitution pair is defined by:*

$$\varphi : \mathbf{SS} \to \mathbf{H} : \varphi(\sigma, \rho) := \{\, (\mathcal{C}(\sigma, x), \mathcal{C}(\rho, x)) \mid x \in \mathrm{dom}(\sigma) \,\}$$

Lemma 6.4.8 *If $\sigma \sim \rho$ then $\varphi(\sigma, \rho)$ is consistent.*

Proof. Take any $(M, N) \in \varphi(\sigma, \rho)$ and $x \in \mathrm{dom}(\sigma)$ such that $M = \mathcal{C}(\sigma, x)$ and $N = \mathcal{C}(\rho, x)$.

1. By condition 1 for $\sigma \sim \rho$ we have that $M \in \mathcal{N}$ iff $N \in \mathcal{N}$.
2. Take any $(M', N') \in \varphi(\sigma, \rho)$ and $x' \in \mathrm{dom}(\sigma)$ such that $M' = \mathcal{C}(\sigma, x')$ and $N' = \mathcal{C}(\rho, x')$. By condition 2 for $\sigma \sim \rho$ we have that $M = M'$ iff $N = N'$.
3. Since $M = \mathcal{C}(\sigma, x) \in \mathcal{I}(\sigma)$ it can not be further decrypted by any key $a \in \mathcal{A}(\sigma)$. By the definition of $\mathcal{I}(\cdot)$, $a \in \mathcal{A}(\sigma)$ iff $a \in \mathcal{I}(\sigma) = \{\mathcal{C}(\sigma, x) | x \in \mathrm{dom}(\sigma)\} = \pi_1(\varphi(\sigma, \rho))$. A symmetrical argument holds for $\mathcal{C}(\rho, x_i)$.

□

Corollary 6.4.9 $\mathrm{fn}(\sigma) = \mathrm{n}(\pi_1(\varphi(\sigma, \rho)))$ *and* $\mathrm{fn}(\rho) = \mathrm{n}(\pi_2(\varphi(\sigma, \rho)))$

An alternative definition of $\varphi(\sigma, \rho)$ is derived from the following lemma.

Lemma 6.4.10 *Let $\sigma \sim \rho$ and $h = \{\, (\sigma(x_i), \rho(x_i)) \mid x_i \in \mathrm{dom}(\sigma) \,\}$. Then $\varphi(\sigma, \rho) = \mathcal{I}(h)$.*

Proof.

1. First we show that whenever $M \in \mathcal{A}(\sigma)$ there are $i, \tilde{\iota}$ such that $M = \mathrm{E}_{\tilde{\iota}}(\sigma_i)$, $\mathrm{E}_{\tilde{\iota}}(\rho_i) \in \mathcal{A}(\rho)$ and $(M, \mathrm{E}_{\tilde{\iota}}(\rho_i)) \in \mathcal{A}(h)$. We use induction on the derivation of $M \in \mathcal{A}(\sigma)$.
   If $M = \sigma(x_i)$ then we get $i, \tilde{\iota}$ such that $M = \mathrm{E}_{\tilde{\iota}}(\sigma_i)$ and $\mathrm{E}_{\tilde{\iota}}(\rho_i) = \rho(x_i)$ by condition 3 of $\sigma \sim \rho$. By definition, $(M, \mathrm{E}_{\tilde{\iota}}(\rho_i)) \in h \subseteq \mathcal{A}(h)$
   Else, there is a such that $\mathrm{E}_a(M), a \in \mathcal{A}(\sigma)$. By the induction assumption there are $i, \tilde{\iota}$ such that $\mathrm{E}_a(M) = \mathrm{E}_{\tilde{\iota}}(\sigma_i)$, $N = \mathrm{E}_{\tilde{\iota}}(\rho_i)$, $(\mathrm{E}_a(M), N) \in \mathcal{A}(h)$ and $(a, \mathcal{C}(\rho, x_{\iota_m})) \in \mathcal{A}(h)$ where $m = |\tilde{\iota}|$. If $\tilde{\iota}' = \iota_1\iota_2 \ldots \iota_{m-1}$ then $M = \mathrm{E}_{\tilde{\iota}'}(\sigma_i)$ and using HEDGE-DEC we get that $(M, \mathrm{E}_{\tilde{\iota}'}(\rho_i)) \in \mathcal{A}(h)$. As $\mathcal{C}(\rho, x_{\iota_m}) \in \mathcal{A}(\rho)$ we have $\mathrm{E}_{\tilde{\iota}'}(\rho_i) \in \mathcal{A}(\rho)$ by SET-DEC.
2. Secondly, we show that if $(M, N) \in \mathcal{A}(h)$ then there are $i, \tilde{\iota}$ such that $M = \mathrm{E}_{\tilde{\iota}}(\sigma_i) \in \mathcal{A}(\sigma)$ and $N = \mathrm{E}_{\tilde{\iota}}(\rho_i) \in \mathcal{A}(\rho)$ by induction on the derivation of $(M, N) \in \mathcal{A}(h)$. The base case follows from condition 3 of $\sigma \sim \rho$.
   Otherwise, there are a, b such that $(a, b) \in \mathcal{A}(h)$ and $(\mathrm{E}_a(M), \mathrm{E}_b(N)) \in \mathcal{A}(h)$. By the induction assumption there are $j, i, \tilde{\iota}$ with $a = \mathcal{C}(\sigma, x_j) \in \mathcal{A}(\sigma)$, $b = \mathcal{C}(\rho, x_j) \in \mathcal{A}(\rho)$, $\mathrm{E}_a(M) = \mathrm{E}_{\tilde{\iota}}(\sigma_i) \in \mathcal{A}(\sigma)$ and $\mathrm{E}_b(N) = \mathrm{E}_{\tilde{\iota}}(\rho_i) \in \mathcal{A}(\rho)$. By SET-DEC $M \in \mathcal{A}(\sigma), N \in \mathcal{A}(\rho)$. Clearly $M = \mathrm{E}_{\tilde{\iota}'}(\sigma_i)$ and $N = \mathrm{E}_{\tilde{\iota}'}(\rho_i)$, where $\tilde{\iota}' = \iota_1\iota_2 \ldots \iota_{m-1}, m = |\tilde{\iota}|$.

By 2, if $(M, N) \in \mathcal{A}(h)$ then there are $i, \tilde{\iota}$ such that $M = \mathrm{E}_{\tilde{\iota}}(\sigma_i)$ and $N = \mathrm{E}_{\tilde{\iota}}(\rho_i)$. If $|\tilde{\iota}| = m > 0$, we have by 1 that $(\mathcal{C}(\sigma, x_{\iota_m}), \mathcal{C}(\rho, x_{\iota_m})) \in \mathcal{A}(h)$, so $(M, N) \notin \mathcal{I}(h)$. Using 1 we also have that $\forall x_i \in \mathrm{dom}(\sigma) : (\mathcal{C}(\sigma, x_i), \mathcal{C}(\rho, x_i)) \in \mathcal{I}(h)$. □

From Hedges to Alleys

For the transformation of hedges into substitution pairs, we have to invent the domain for the substitutions. However, it does not matter which particular domain we select, since the domain of the substitutions is not important for the definition of alleyed bisimulation. Note that $\mathcal{M} \times \mathcal{M}$ and $\mathcal{N}$ are both countably infinite, so there is a bijection between them. To invent the substitution domain, we may use any such bijection.

Definition 6.4.11 *Fix a bijection $f : \mathcal{M} \times \mathcal{M} \to \mathcal{N}$. The alley corresponding to a hedge is defined by:*

$$\begin{array}{rrcl} \theta^f : \mathbf{H} \to \mathbf{SS} : & \theta^f(h) & := & (\, \theta_1^f(h) \,,\, \theta_2^f(h) \,) \\ & \theta_1^f(h) & := & \{\, \{^{M}/_{f(M,N)}\} \mid (M,N) \in h \,\} \\ & \theta_1^f(h) & := & \{\, \{^{N}/_{f(M,N)}\} \mid (M,N) \in h \,\} \end{array}$$

We sometimes use the projections $h_1^f := \theta_1^f(h)$ and $h_2^f := \theta_2^f(h)$to denote the left and right substitutions corresponding to h (under f).

In the following, we implicitly assume a suitable fixed bijection f.

Lemma 6.4.12 *If h is consistent then $h_1^f \sim h_2^f$ and $h = \varphi(\theta^f(h))$.*

Proof. Clearly h_1^f and h_2^f have the same domain. By condition 3 for the consistency of h we have that $h_1^f(x) = \mathcal{C}(h_1^f, x)$ and $h_2^f(x_i) = \mathcal{C}(h_2^f, x_i)$, so $h = \{\, (\mathcal{C}(h_1^f, x_i), \mathcal{C}(h_2^f, x_i)) \mid x_i \in \mathrm{dom}(h_1^f) \,\}$ as desired.

To show that $h_1^f \sim h_2^f$ we fix $x \in \mathrm{dom}(h_1^f)$.

1. $\mathcal{C}(h_1^f, x) \in \mathcal{N}$ iff $\mathcal{C}(h_2^f, x) \in \mathcal{N}$ by condition 1 of the consistency of h.
2. If $y \in \mathrm{dom}(h_1^f)$ then $\mathcal{C}(h_1^f, x) = \mathcal{C}(h_1^f, y)$ and $\mathcal{C}(h_2^f, x) = \mathcal{C}(h_2^f, y)$ iff $x = y$ by condition 2 of the consistency of h.
3. As above, $h_1^f(x) = \mathcal{C}(h_1^f, x)$ and $h_2^f(x) = \mathcal{C}(h_2^f, x)$.

□

Thus, θ^f preserves consistency. Also, since φ is an inverse of θ^f we get that if φ preserves the synthesis, then so does θ^f. We now proceed to show that it is indeed true that φ preserves environment synthesis.

Sending and Receiving Messages

In this paragraph we always implicitly assume that $\sigma \sim \rho$ and that $h = \varphi(\sigma, \rho)$.

The following two lemmas show that the hedge derived from a consistent pair of substitutions can generate exactly the same message pairs as the substitutions. In other words, φ preserves the synthesis of environments.

Lemma 6.4.13 *If $(\sigma, \rho) \vdash M \leftrightarrow N$ then $h \vdash M \leftrightarrow N$.*

Proof. By the definition of $(\sigma, \rho) \vdash M \leftrightarrow N$ we get that there exits ζ such that $\mathrm{n}(\zeta) \subseteq \mathrm{dom}(\sigma)$, $M = [\![\, \zeta\sigma \,]\!]$ and $N = [\![\, \zeta\rho \,]\!]$.

By Lemma 6.4.2 we get that $([\![\, \zeta\sigma \,]\!], [\![\, \zeta\rho \,]\!]) = (\mathrm{E}_{\tilde{\iota}}(\sigma_i), \mathrm{E}_{\tilde{\iota}}(\rho_i))$ for some $i, \tilde{\iota}$. We prove that $h \vdash [\![\, \zeta\sigma \,]\!] \leftrightarrow [\![\, \zeta\rho \,]\!]$ by induction on $|\tilde{\iota}|$.

- If $|\tilde{\iota}| = 0$, then $([\![\, \zeta\sigma \,]\!], [\![\, \zeta\rho \,]\!]) = (\mathcal{C}(\sigma, x_i), \mathcal{C}(\rho, x_i)) \in h$.
- Else, if $|\tilde{\iota}| = m + 1$ and $\tilde{\iota}' = \iota_1\iota_2 \ldots \iota_m$ then $h \vdash \mathrm{E}_{\tilde{\iota}'}(\sigma_i) \leftrightarrow \mathrm{E}_{\tilde{\iota}'}(\rho_i)$ by the induction assumption. As $(\mathcal{C}(\sigma, x_{\iota_{m+1}}), \mathcal{C}(\rho, x_{\iota_{m+1}})) \in h$ we can use EQ-ENC to derive that $h \vdash \mathrm{E}_{\tilde{\iota}}(\sigma_i) \leftrightarrow \mathrm{E}_{\tilde{\iota}}(\rho_i)$.

□

Lemma 6.4.14 *If $h \vdash M \leftrightarrow N$ then $(\sigma, \rho) \vdash M \leftrightarrow N$*

Proof. The proof is by induction on the derivation of $h \vdash M \leftrightarrow N$. First note that for each $x_i \in \mathrm{dom}(\sigma)$ we can by Lemma 6.4.1 find an η_i such that $\mathrm{n}(\eta_i) \subseteq \mathrm{dom}(\sigma) \wedge [\![\, \eta_i\sigma \,]\!] = \mathcal{C}(\sigma, x_i)$. By Lemma 6.4.2, $([\![\, \eta_i\sigma \,]\!], [\![\, \eta_i\rho \,]\!]) = (\mathcal{C}(\sigma, x_i), \mathcal{C}(\rho, x_i))$.

- If $(M, N) \in h$ then there is a i such that $(M, N) = (\mathcal{C}(\sigma, x_i), \mathcal{C}(\rho, x_i)) = ([\![\, \eta_i\sigma \,]\!], [\![\, \eta_i\rho \,]\!])$.

- Else, $(M,N) = (\mathrm{E}_a(M'), \mathrm{E}_b(N'))$ and by the induction assumption there exists ζ such that $(M',N') = (\llbracket \zeta\sigma \rrbracket, \llbracket \zeta\rho \rrbracket)$. By Lemma 6.1.1, $(a,b) \in h$, so there exists i such that $(a,b) = (\mathcal{C}(\sigma, x_i), \mathcal{C}(\rho, x_i)) = (\llbracket \eta_i\sigma \rrbracket, \llbracket \eta_i\rho \rrbracket)$. But then $(M,N) = (\mathrm{E}_{\llbracket \eta_i\sigma \rrbracket}(\llbracket \zeta\sigma \rrbracket), \mathrm{E}_{\llbracket \eta_i\rho \rrbracket}(\llbracket \zeta\rho \rrbracket)) = (\llbracket \mathrm{E}_{\eta_i}(\zeta)\,\sigma \rrbracket, \llbracket \mathrm{E}_{\eta_i}(\zeta)\,\rho \rrbracket)$.

□

We now show that an environment represented by a consistent pair of substitutions can accept a given message pair iff an environment represented by a hedge can do it, and characterize the hedge derived under φ.

Lemma 6.4.15 *If $\sigma\{^{M}/_{x}\} \sim \rho\{^{N}/_{x}\}$ then $\mathcal{I}(h \cup \{(M,N)\}) = \varphi(\sigma\{^{M}/_{x}\}, \rho\{^{N}/_{x}\})$.*

Proof. Let $g = \{(\sigma(x_i), \rho(x_i)) | x_i \in \mathrm{dom}(\sigma)\}$. By Lemma 6.1.15, $\mathcal{I}(\mathcal{I}(g) \cup \{(M,N)\}) = \mathcal{I}(g \cup \{(M,N)\})$. By using Lemma 6.4.10 to (σ,ρ) we get that $\mathcal{I}(g) = h$, and substitution gives that $\mathcal{I}(h \cup \{(M,N)\}) = \mathcal{I}(g \cup \{(M,N)\})$. Applying Lemma 6.4.10 to $(\sigma\{^{M}/_{x}\}, \rho\{^{N}/_{x}\})$ we get $\mathcal{I}(g \cup \{(M,N)\}) = \varphi(\sigma\{^{M}/_{x}\}, \rho\{^{N}/_{x}\})$. Thus $\mathcal{I}(h \cup \{(M,N)\}) = \varphi(\sigma\{^{M}/_{x}\}, \rho\{^{N}/_{x}\})$, as desired. □

Lemma 6.4.16 *If $\mathcal{I}(h \cup \{(M,N)\})$ is consistent, then $\sigma\{^{M}/_{x}\} \sim \rho\{^{N}/_{x}\}$ for any $x \notin \mathrm{dom}(\sigma)$.*

Proof. Let $g = \mathcal{I}(h \cup \{(M,N)\})$. By Lemma 6.4.12 $g_1^f \sim g_2^f$. If $\mathrm{dom}(g_1^f) \cap (\mathrm{dom}(\sigma) \cup \{x\}) \neq \emptyset$ we first use Lemma 6.4.3 and Lemma 6.4.4 to change the domain of g_i^f, which is equivalent to choosing a different bijection f.

By Lemma 6.4.13 we have that $\forall x_i \in \mathrm{dom}(\sigma) : h \vdash \sigma(x_i) \leftrightarrow \rho(x_i)$. By Lemma 6.1.16 we have that $h \cup \{(M,N)\} \leq g$, so $g \vdash \sigma(x_i) \leftrightarrow \rho(x_i)$. By Lemma 6.4.14 there are η_i such that $\llbracket \eta_i g_1^f \rrbracket = \sigma(x_i), \llbracket \eta_i g_2^f \rrbracket = \rho(x_i)$ and $\mathrm{n}(\eta_i) \subseteq \mathrm{dom}(g_1^f)$.

Then, by iterated application of Lemma 6.4.3 we get that $g_1^f \circ \sigma \sim g_2^f \circ \rho$. Similarly, as $g \vdash M \leftrightarrow N$ there is, by Lemma 6.4.14, a ζ such that $\mathrm{n}(\zeta) \subseteq \mathrm{dom}(g_1^f)$ and $\llbracket \zeta g_1^f \rrbracket = M, \llbracket \zeta g_2^f \rrbracket = N$. By Lemma 6.4.3 $g_1^f \circ \sigma\{^{M}/_{x}\} \sim g_2^f \circ \rho\{^{N}/_{x}\}$. Finally, by Lemma 6.4.4 $\sigma\{^{M}/_{x}\} \sim \rho\{^{N}/_{x}\}$. □

We also need to see what happens when the environment creates fresh names.

Lemma 6.4.17 *If $B = \{b_1, b_2, \ldots, b_n\}$ is a finite set of names such that $B \cap \mathrm{fn}(\sigma,\rho) = \emptyset$ then $h \cup \mathrm{Id}_B = \varphi(\sigma\{^{b_1}/_{c_1}, \ldots, ^{b_n}/_{c_n}\}, \rho\{^{b_1}/_{c_1}, \ldots, ^{b_n}/_{c_n}\})$ for all sets of names $C = \{c_1, c_2, \ldots, c_n\}$ such that $C \cap \mathrm{dom}(\sigma) = \emptyset$.*

Proof. We write $\{^{B}/_{C}\}$ for $\{^{b_1}/_{c_1}, \ldots, ^{b_n}/_{c_n}\}$. By Lemma 6.4.8 we have that h is consistent, so since B is fresh we have that $h \cup \mathrm{Id}_B$ is consistent by Lemma 6.1.20. By Lemma 6.4.16 we get that $\sigma\{^{B}/_{C}\} \sim \rho\{^{B}/_{C}\}$.

Since $b_i \in \mathcal{N}$ we have that $\mathcal{C}(\sigma\{^{B}/_{C}\}, b_i) = b_i$ and $\mathcal{C}(\rho\{^{B}/_{C}\}, b_i) = b_i$. As B is fresh we have for any $x_i \in \mathrm{dom}(\sigma)$ that $\mathcal{C}(\sigma\{^{B}/_{C}\}, x_i) = \mathcal{C}(\sigma, x_i)$ and $\mathcal{C}(\rho\{^{B}/_{C}\}, x_i) = \mathcal{C}(\rho, x_i)$, so $\sigma\{^{B}/_{C}\} \sim \rho\{^{B}/_{C}\}$ and $\varphi(\sigma\{^{B}/_{C}\}, \rho\{^{B}/_{C}\}) = h \cup \mathrm{Id}_B$. □

We use the preceeding result to show a variant of Lemma 6.4.14, namely that if a hedge can create a pair of new messages containing some fresh names then the same pair of messages can be produced by a consistent pair of substitutions.

Lemma 6.4.18 *If B is a finite set of names such that $B \cap \mathrm{fn}(\sigma,\rho) = \emptyset$ and $h \cup \mathrm{Id}_B \vdash M \leftrightarrow N$ then there exists η such that $\mathrm{n}(\eta) \setminus \mathrm{dom}(\sigma) = B$ and $\llbracket \eta\sigma \rrbracket = M, \llbracket \eta\rho \rrbracket = N$.*

Proof. $h \cup \mathrm{Id}_B = \varphi(\sigma\{^{b_1}/_{c_1}, \ldots, ^{b_n}/_{c_n}\}, \rho\{^{b_1}/_{c_1}, \ldots, ^{b_n}/_{c_n}\})$ by Lemma 6.4.17. Then, according to Lemma 6.4.14 there is $\zeta : \mathrm{n}(\zeta) \subseteq \mathrm{dom}(\sigma) \cup B$ and $\llbracket \zeta\sigma \rrbracket = M, \llbracket \zeta\rho \rrbracket = N$. Clearly, $\eta = \mathrm{D}_{b_1}(\mathrm{D}_{b_2}(\cdots \mathrm{D}_{b_n}(\mathrm{E}_{b_n}(\cdots \mathrm{E}_{b_2}(\mathrm{E}_{b_1}(\zeta))\cdots))\cdots))$ has the desired properties. □

6.5 Frames/Fences vs. Alleys

The missing component in our relations between the various kinds of environments concerns the relation between frame-theory pairs and substitution pairs. As a matter of fact, we are only interested in the case of strongly consistent substitution pairs.

Definition 6.5.1 *The frame-theory pair corresponding to a (strongly consistent) substitution pair is defined by:*

$$\gamma : \mathbf{SS} \rightarrow \mathbf{FT} : \gamma(\sigma,\rho) := \{ \, (\, \mathcal{N} \cap \pi_1(\varphi(\sigma,\rho)) \, , \, \varphi(\sigma,\rho) \setminus (\mathcal{N} \times \mathcal{N}) \,) \, \}$$

Note that this definition indeed only makes sense in the case that $\sigma \sim_{\mathrm{s}} \rho$, because in it we consider for the frame only the names of $\pi_1(\varphi(\sigma,\rho))$ while ignoring those of $\pi_2(\varphi(\sigma,\rho))$. Note further that all of the properties we need later on for γ will be derived from the properties of φ.

6.6 Message Equivalence

As we saw in Section 5, two consistent non-blind environments can be $\approx^{\mathrm{x}}_{\mathrm{y}}$-equivalent only if they are $\mathcal{M}$-equivalent (see Definition 5.4.1). With the results of this section, we can fully characterize $\mathcal{M}$-equivalent frame-theory pairs and alleys and, because of this, give necessary conditions on full abstractions on the corresponding bisimilarities.

As showed in Lemma 6.1.18, two $\mathcal{M}$-equivalent consistent hedges are equal. Using this, we can show that $\mathcal{M}$-equivalent consistent frame-theory pairs are equal. First we need to show that ψ is injective when restricted to the set of consistent frame-theory pairs.

Lemma 6.6.1 *If* $(\mathrm{fr},\mathrm{th})$ *and* $(\mathrm{fr}',\mathrm{th}')$ *are consistent and* $\psi(\mathrm{fr},\mathrm{th}) = \psi(\mathrm{fr}',\mathrm{th}')$ *then* $(\mathrm{fr},\mathrm{th}) = (\mathrm{fr}',\mathrm{th}')$.

Proof. Let $h = \psi(\mathrm{fr},\mathrm{th}) = \psi(\mathrm{fr}',\mathrm{th}')$. By definition, $(M,N) \in h$ iff $M = N \in \mathrm{fr}$, $M = N \in \mathrm{fr}'$, $(M,N) \in \mathrm{th}$ or $(M,N) \in \mathrm{th}'$.

Take $(M,N) \in h$. If M or N is a name then $M = N \in \mathrm{fr}$ and $M = N \in \mathrm{fr}'$ by Lemma 6.2.2. Otherwise we must have $(M,N) \in \mathrm{th}$ and $(M,N) \in \mathrm{th}'$, since the frame only contains names. □

Two $\mathcal{M}$-equivalent consistent frame-theory pairs are equal.

Lemma 6.6.2 *If* $(\mathrm{fr},\mathrm{th})$ *and* $(\mathrm{fr}',\mathrm{th}')$ *are consistent and* $(\mathrm{fr},\mathrm{th}) \gtrless (\mathrm{fr}',\mathrm{th}')$ *then* $(\mathrm{fr},\mathrm{th}) = (\mathrm{fr}',\mathrm{th}')$.

Proof. Note that $\psi(\mathrm{fr},\mathrm{th}) \gtrless \psi(\mathrm{fr}',\mathrm{th}')$. $\psi(\mathrm{fr},\mathrm{th})$ and $\psi(\mathrm{fr}',\mathrm{th}')$ are both consistent by Lemma 6.2.3, so $\psi(\mathrm{fr},\mathrm{th}) = \psi(\mathrm{fr}',\mathrm{th}')$ by Lemma 6.1.18. Then $(\mathrm{fr},\mathrm{th}) = (\mathrm{fr}',\mathrm{th}')$ by Lemma 6.6.1. □

Corollary 6.6.3 *Together with Proposition 5.4.3, this gives that all full abstractions from* [*framed, fenced*] *to* [*framed, fenced*] *must be the identity on consistent, non-blind environments.*

$\mathcal{M}$-equivalent pairs of substitutions are exactly those that correspond to the same hedge.

Lemma 6.6.4 $(\sigma,\rho) \gtrless (\sigma',\rho')$ *if and only if* $\varphi(\sigma,\rho) = \varphi(\sigma',\rho')$.

Proof. By Lemma 6.4.13 and Lemma 6.4.14 we have that $(\sigma,\rho) \gtrless (\sigma',\rho')$ iff $\varphi(\sigma,\rho) \gtrless \varphi(\sigma',\rho')$. $\varphi(\sigma,\rho)$ and $\varphi(\sigma',\rho')$ are consistent by Lemma 6.4.8, so Lemma 6.1.18 gives that $\varphi(\sigma,\rho) \gtrless \varphi(\sigma',\rho')$ iff $\varphi(\sigma,\rho) = \varphi(\sigma',\rho')$. □

Corollary 6.6.5 *Together with Proposition 5.4.3, this gives that all full abstractions from* [*alley, trellis*] *to* [*alley, trellis*] *must preserve the image under* φ *of all consistent, non-blind environments.*

7 Comparing Bisimulations

Having established the relations between the different kinds of environment representations, we may study the relations between the respective bisimulations. In general, this is done by lifting the environment mapping functions to consistent environmental relations. In general the lifiting is defined as follows:

Assume that $g : \mathbf{E}_\mathrm{x} \to \mathbf{E}_\mathrm{y}$ is an environment mapping.
If e_x is consistent for $\approx_\mathrm{x}$ we let $G(e_\mathrm{x}, P, Q) := (g(e_\mathrm{x}), P, Q)$.
If $\mathcal{R}$ is a consistent relation (for $\approx_\mathrm{x}$) we let $G(\mathcal{R}) := \{\, G(e_\mathrm{x}, P, Q) \mid e_\mathrm{x} \vdash P \mathrel{\mathcal{R}} Q \,\}$.
If $\mathcal{S}$ is a consistent relation (for $\approx_\mathrm{y}$) we let $G^{-1}(\mathcal{S}) := \{(e_\mathrm{x}, P, Q) \mid G(e_\mathrm{x}, P, Q) \in \mathcal{S}\}$.

7.1 Fenced vs. Trellis Bisimulation

These two bisimulations were compared by [FHJ01]. Here, we recapitulate their results. We use the environment mapping γ and its lifting Γ, as instances of the above definition scheme.

These are the main results of [FHJ01]:

Theorem 7.1.1 $\Gamma(\approx_\mathrm{s})$ *is a fenced bisimulation.*

Theorem 7.1.2 $\Gamma^{-1}(\approx_\#)$ *is a strongly consistent alley bisimulation.*

Using the terminology of Definition 5.2.1, these results give us the following Lemma.

Lemma 7.1.3 $\approx_\#$ *is fully γ-abstract w.r.t.* $\approx_\mathrm{s}$.

Proof. Fix $\sigma \sim_\mathrm{s} \rho$. We need to check that for all processes P, Q we have that $(\sigma, \rho) \vdash P \approx_\mathrm{s} Q$ if and only if $\gamma(\sigma, \rho) \vdash P \approx_\# Q$. By Theorem 7.1.2 we have the "if" and Theorem 7.1.1 gives the "only if". □

Intuitively, this means that $\approx_\mathrm{s}$ can be embedded in $\approx_\#$. However, this is only one of the full abstractions needed to prove fenced and trellis equivalent. For the other direction, we must find an encironment mapping going in the opposite direction of γ. The composition of ψ and θ^f is a reasonable candidate.

Lemma 7.1.4 *1. If* $(\mathrm{fr}, \mathrm{th})$ *is consistent then* $\theta^f_1(\psi(\mathrm{fr}, \mathrm{th})) \sim_\mathrm{s} \theta^f_2(\psi(\mathrm{fr}, \mathrm{th}))$.
2. $\gamma \circ \theta^f \circ \psi = \mathrm{Id}$ *on the set of consistent frame-theory pairs.*

Proof. $\psi(\mathrm{fr}, \mathrm{th})$ is consistent by Lemma 6.2.3, so $\theta^f_1(\psi(\mathrm{fr}, \mathrm{th})) \sim \theta^f_1(\psi(\mathrm{fr}, \mathrm{th}))$ according to Lemma 6.4.12. Since $(\mathrm{fr}, \mathrm{th})$ is consistent, the only pairs of names in $\psi(\mathrm{fr}, \mathrm{th})$ are of the type (a, a) where $a \in \mathrm{fr}$. This gives that we actually have $\theta^f_1(\psi(\mathrm{fr}, \mathrm{th})) \sim_\mathrm{s} \theta^f_1(\psi(\mathrm{fr}, \mathrm{th}))$.

By Lemma 6.4.12 we also have that $\varphi(\theta^f(\psi(\mathrm{fr}, \mathrm{th}))) = \psi(\mathrm{fr}, \mathrm{th})$. Since $(\mathrm{fr}, \mathrm{th})$ is consistent by assumption, we have that $\mathrm{fr} = \mathcal{N} \cap \pi_1(\psi(\mathrm{fr}, \mathrm{th}))$ and $\mathrm{th} = \psi(\mathrm{fr}, \mathrm{th}) \setminus (\mathcal{N} \times \mathcal{N})$. □

Lemma 7.1.5 $\approx_\mathrm{s}$ *is fully $\theta^f \circ \psi$-abstract w.r.t.* $\approx_\#$.

Proof. Fix a consistent frame-theory pair $(\mathrm{fr}, \mathrm{th})$. We need to check that for all processes P, Q we have that $(\mathrm{fr}, \mathrm{th}) \vdash P \approx_\# Q$ iff $\theta^f(\psi(\mathrm{fr}, \mathrm{th})) \vdash P \approx_\mathrm{s} Q$.

We have by Lemma 7.1.4 that $\Gamma(\theta^f(\psi(\mathrm{fr}, \mathrm{th})), P, Q) = ((\mathrm{fr}, \mathrm{th}), P, Q)$. Using this, Theorem 7.1.1 gives the "if" and the "only if" follows from Theorem 7.1.2. □

By combining Lemma 7.1.5 and Lemma 7.1.3, we have the desired equivalence.

Theorem 7.1.6 $\approx_\mathrm{s}$ *is $(\gamma, \theta^f \circ \psi)$-equivalent to* $\approx_\#$.

7.2 Fenced vs. Hedged Bisimulation

The correspondence between fenced and hedged bisimulation is easy to find, since they have similar environments and, according to Lemma 6.3.3, the same action on process output. On the level of the environments the correspondence is given by ψ, and we lift this to Ψ according to the above definition scheme.

We immediately get an embedding of fenced bisimulation in hedged.

Theorem 7.2.1 *$\Psi(\approx_{\#})$ is a hedged bisimulation.*

Proof. $\Psi(\approx_{\#})$ is a consistent hedged relation by Lemma 6.2.3 and is symmetric by the symmetry of $\approx_{\#}$. Assume that $(\mathrm{fr}, \mathrm{th}) \vdash P \approx_{\#} Q$.

1. If $P \xrightarrow{\tau} P'$ then there is a Q' such that $Q \Rightarrow Q'$ and $(\mathrm{fr}, \mathrm{th}) \vdash P' \approx_{\#} Q'$, so $(\psi(\mathrm{fr}, \mathrm{th}), P', Q') \in \Psi(\approx_{\#})$.
2. Assume that $P \xrightarrow{a(M)} P'$, $\psi(\mathrm{fr}, \mathrm{th}) \vdash a \leftrightarrow b$ and $B \subset \mathcal{N}$ is finite such that $B \cap (\mathrm{fn}(P, Q) \cup \mathrm{n}(\psi(\mathrm{fr}, \mathrm{th}))) = \emptyset$ and $\psi(\mathrm{fr}, \mathrm{th}) \cup \mathrm{Id}_B \vdash M \leftrightarrow N$. By Lemma 6.1.1 $(a, b) \in \psi(\mathrm{fr}, \mathrm{th})$ and as $(\mathrm{fr}, \mathrm{th})$ is consistent we have that $a = b \in \mathrm{fr}$. Clearly $\mathrm{n}(\psi(\mathrm{fr}, \mathrm{th})) = \mathrm{fr} \cup \mathrm{n}(\mathrm{th})$. Then there exists a Q' such that $Q \overset{a(N)}{\Longrightarrow} Q'$ and $(\mathrm{fr} \cup B, \mathrm{th}) \vdash P' \approx_{\#} Q'$, so $(\psi(\mathrm{fr}, \mathrm{th}) \cup \mathrm{Id}_B, P', Q') \in \Psi(\approx_{\#})$.
3. Assume that $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $\psi(\mathrm{fr}, \mathrm{th}) \vdash a \leftrightarrow b$ and $\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{n}(\pi_1(h))) = \emptyset$. As above $a = b \in \mathrm{fr}$ and $\mathrm{n}(\pi_1(\psi(\mathrm{fr}, \mathrm{th}))) = \mathrm{fr} \cup \mathrm{n}(\pi_1(\mathrm{th}))$ so there are $Q', N, \tilde{d}$ such that $Q \overset{(\nu\tilde{d})\,\overline{b}\,N}{\Longrightarrow} Q'$, $(\mathrm{fn}(Q) \cup \mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th}))) \cap \{\tilde{d}\} = \emptyset$ and $\xi(\mathrm{fr}, \mathrm{th}, M, N) \vdash P' \approx_{\#} Q'$. Clearly $\mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th})) = \mathrm{n}(\pi_2(\psi(\mathrm{fr}, \mathrm{th})))$. $\psi(\xi(\mathrm{fr}, \mathrm{th}, M, N)) = \mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\})$ by Lemma 6.3.3, so $(\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}), P', Q') \in \Psi(\approx_{\#})$.

□

We may also state this as

Corollary 7.2.2 *$\approx_{\mathrm{h}}$ is ψ-complete w.r.t. $\approx_{\#}$.*

7.3 Fenced vs. Framed Bisimulation

One of the main results of [EHHO99] is the following:

Theorem 7.3.1 *If $(\mathrm{fr}, \mathrm{th}) \vdash P \approx_{\#} Q$ then $(\mathrm{fr}, \mathrm{th}) \vdash P \approx_{\mathrm{f}} Q$.*

The authors enunciate that the converse would also hold, but our counterexamples in Section 4 show that this cannot be the case. Indeed, their proof is flawed[3]. However, Theorem 7.3.1 can be restated as

Corollary 7.3.2
- *$\approx_{\mathrm{f}}$ is Id-complete w.r.t. $\approx_{\#}$.*
- *$\approx_{\#}$ is Id-sound w.r.t. $\approx_{\mathrm{f}}$.*

By transitivity, we also get

Corollary 7.3.3
- *$\approx_{\mathrm{f}}$ is γ-complete w.r.t. $\approx_{\mathrm{s}}$.*
- *$\approx_{\mathrm{s}}$ is $\theta^f \circ \psi$-sound w.r.t. $\approx_{\mathrm{f}}$.*

[3] In the proof of their Lemma 20, stating (roughly) soundness of framed up to weakening, actions that must be simulated by the minimal environment but not by a larger one are not taken into account (cf the example in Section 4).

7.4 Framed vs. Hedged Bisimulation

In framed bisimulation we can make arbitrary extensions to the environment on process output, something that is not permitted in hedged bisimulation. Therefore, we must show that extending a hedge is sound with respect to hedged bisimulation, i.e., that the extended hedge does not relate any process pairs not related by the original hedge.

We show this by transferring a corresponding result for alley bisimulation to hedged bisimulation. To do this we use Lemma 7.5.3, which will be proven (independently of this comparison between framed and hedged bisimulation) in Section 7.5. We use this forward reference to keep together the comparisons between framed, fenced and hedged bisimilarities, hopefully making this part of the paper easier rather than harder to follow.

First two results by [BDP02]:

Lemma 7.4.1 *If* $\mathrm{n}(\zeta) \subseteq \mathrm{dom}(\sigma)$*,* $[\![\zeta\sigma]\!] \neq \perp$ *and* $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$ *then* $(\sigma\{{}^{[\![\zeta\sigma]\!]}/_{x}\}, \rho\{{}^{[\![\zeta\rho]\!]}/_{x}\}) \vdash P \approx_{\mathrm{a}} Q$*.*

Lemma 7.4.2 *If* $(\sigma\{{}^{M}/_{x}\}, \rho\{{}^{N}/_{x}\}) \vdash P \approx_{\mathrm{a}} Q$ *then* $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$*.*

For reference, we also state Lemma 7.5.3.

Lemma 7.4.3 (Lemma 7.5.3) $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$ *iff* $\varphi(\sigma, \rho) \vdash P \approx_{\mathrm{h}} Q$*.*

We use these to show that hedged bisimulation is sound up to weakening.

Theorem 7.4.4 *Let* $g \leq h$ *be consistent. If* $h \vdash P \approx_{\mathrm{h}} Q$ *then* $g \vdash P \approx_{\mathrm{h}} Q$*.*

Proof. By Lemma 7.4.3 we have that $\varphi(\sigma, \rho) \vdash P \approx_{\mathrm{h}} Q$ implies that $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$. Since $\varphi(\theta^f(h)) = h$ by Lemma 6.4.12, we can apply this to $(\sigma, \rho) = \theta^f(h)$ to obtain $\theta^f(h) \vdash P \approx_{\mathrm{a}} Q$.

For all $(M, N) \in g$ we have that $h \vdash M \leftrightarrow N$, so by Lemma 6.4.14 there is ζ such that $\mathrm{n}(\zeta) \subseteq \mathrm{dom}(h_1^f)$, $[\![\zeta h_1^f]\!] = M$ and $[\![\zeta h_2^f]\!] = N$. By iterated application of Lemma 7.4.1 this gives that $(h_1^f \cup g_1^f, h_2^f \cup g_2^f) \vdash P \approx_{\mathrm{a}} Q$.

Using Lemma 7.4.2 we then have that $\theta^f(g) \vdash P \approx_{\mathrm{a}} Q$. By Lemma 6.4.12 we also have that $\varphi(\theta^f(g)) = g$. We get by Lemma 7.4.3 that $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$ implies that $\varphi(\sigma, \rho) \vdash P \approx_{\mathrm{h}} Q$. Instantiating this with $(\sigma, \rho) = \theta^f(g)$ we get the desired $g \vdash P \approx_{\mathrm{h}} Q$. □

We also need a lemma concerning extensions of frame-theory pairs and their corresponding hedges.

Lemma 7.4.5 *If* $(\mathrm{fr}', \mathrm{th}')$ *is consistent,* $(\mathrm{fr}, \mathrm{th}) \leq (\mathrm{fr}', \mathrm{th}')$ *and* $(\mathrm{fr}', \mathrm{th}') \vdash M \leftrightarrow N$ *then* $\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\})$ *is consistent and* $\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}) \leq \psi(\mathrm{fr}', \mathrm{th}')$*.*

Proof. By Corollary 6.1.5(2) $\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\} \leq \psi(\mathrm{fr}', \mathrm{th}')$. This gives by Lemma 6.1.13 that $\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}) \leq \mathcal{I}(\psi(\mathrm{fr}', \mathrm{th}'))$. $\psi(\mathrm{fr}', \mathrm{th}')$ is consistent by Lemma 6.2.3, so $\psi(\mathrm{fr}', \mathrm{th}') = \mathcal{I}(\psi(\mathrm{fr}', \mathrm{th}'))$ according to Lemma 6.1.16.

$\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\})$ is irreducible by Corollary 6.1.8, so the consistency follows from Lemma 6.1.19 applied to $\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}) \leq \psi(\mathrm{fr}', \mathrm{th}')$. □

Using this, we have the desired result.

Theorem 7.4.6 *If* $(\mathrm{fr}, \mathrm{th}) \vdash P \approx_{\mathrm{f}} Q$ *then* $\psi(\mathrm{fr}, \mathrm{th}) \vdash P \approx_{\mathrm{h}} Q$*.*

Proof. Assume that $(\mathrm{fr}, \mathrm{th}) \vdash P \approx_{\mathrm{f}} Q$. Internal calculation and input is handled as in the proof of Theorem 7.2.1.

Assume that $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $\psi(\mathrm{fr}, \mathrm{th}) \vdash a \leftrightarrow b$ and $\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{n}(\pi_1(h))) = \emptyset$. By Lemma 6.2.2 $a = b \in \mathrm{fr}$.

Clearly $\mathrm{n}(\pi_1(\psi(\mathrm{fr},\mathrm{th}))) = \mathrm{fr} \cup \mathrm{n}(\pi_1(\mathrm{th}))$ so there are $Q', N, \tilde{d}, \mathrm{fr}', \mathrm{th}'$ such that $Q \xRightarrow{(\nu\tilde{d})\,\bar{b}\,N} Q'$, $(\mathrm{fn}(Q) \cup \mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th}))) \cap \{\tilde{d}\} = \emptyset$, $(\mathrm{fr},\mathrm{th}) \leq (\mathrm{fr}',\mathrm{th}')$, $(\mathrm{fr}',\mathrm{th}') \vdash M \leftrightarrow N$ and $(\mathrm{fr}',\mathrm{th}') \vdash P' \approx_{\mathrm{f}} Q'$. Clearly $\mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th})) = \mathrm{n}(\pi_2(\psi(\mathrm{fr},\mathrm{th})))$.

We need to show that $\mathcal{I}(\psi(\mathrm{fr},\mathrm{th}) \cup \{(M,N)\}) \vdash P' \approx_{\mathrm{h}} Q'$. Note that $\psi(\mathrm{fr}',\mathrm{th}') \vdash P \approx_{\mathrm{h}} Q$ by induction. By Lemma 7.4.5 we have that $\mathcal{I}(\psi(\mathrm{fr},\mathrm{th}) \cup \{(M,N)\})$ is consistent and $\mathcal{I}(\psi(\mathrm{fr},\mathrm{th}) \cup \{(M,N)\}) \leq \psi(\mathrm{fr}',\mathrm{th}')$, so $\mathcal{I}(\psi(\mathrm{fr},\mathrm{th}) \cup \{(M,N)\}) \vdash P' \approx_{\mathrm{h}} Q'$ by Theorem 7.4.4. □

Corollary 7.4.7 $\approx_{\mathrm{h}}$ *is ψ-complete w.r.t.* $\approx_{\mathrm{f}}$.

In proving this theorem, we needed to weaken the hedges on process output. Actually, since fenced bisimilarity is not complete with respect to framed, construction of a non-minimal environment may be *required* to get a framed bisimulation. Since $\approx_{\mathrm{h}}$ only performs minimal extensions, we get that $\Psi(\approx_{\mathrm{f}})$ is not a hedged bisimulation.

7.5 Alley vs. Hedged bisimulation

Now it is time to show the equivalence between alley bisimilarity and hedged bisimilarity. Recall the environment mappings φ (Definition 6.4.7)and θ^f (Definition 6.4.11). We lift φ to Φ according to the above definition scheme.

Theorem 7.5.1 $\Phi(\approx_{\mathrm{a}})$ *is a hedged bisimulation.*

Proof. $\Phi(\approx_{\mathrm{a}})$ is symmetric by the symmetry of $\approx_{\mathrm{a}}$ and consistent by Lemma 6.4.8. Assume that $(\sigma,\rho) \vdash P \approx_{\mathrm{a}} Q$ and that $h = \varphi(\sigma,\rho)$.

1. If $P \xrightarrow{\tau} P'$ there exists Q' such that $Q \Rightarrow Q'$ and $(\sigma,\rho) \vdash P' \approx_{\mathrm{a}} Q'$, so $\Phi((\sigma,\rho),P',Q') = (h,P',Q') \in \Phi(\approx_{\mathrm{a}})$.
2. Assume that $P \xrightarrow{a\,M} P'$, $h \vdash a \leftrightarrow b$ and $B \subset \mathcal{N}$ is finite such that $B \cap (\mathrm{fn}(P,Q) \cup \mathrm{n}(h)) = \emptyset$ and $h \cup \mathrm{Id}_B \vdash M \leftrightarrow N$.
   By Lemma 6.4.14 we get that $(\sigma,\rho) \vdash a \leftrightarrow b$. By Lemma 6.4.18 there is ζ such that $\mathrm{n}(\zeta) \setminus \mathrm{dom}(\sigma) = B$ and $[\![\zeta\sigma]\!] = M, [\![\zeta\rho]\!] = N$. As $(\sigma,\rho) \vdash P \approx_{\mathrm{a}} Q$ there exist $Q', \tilde{c}$ such that $Q \xRightarrow{b\,N} Q'$ and $(\sigma\{^{b_1}/_{c_1},\dots,^{b_n}/_{c_n}\}, \rho\{^{b_1}/_{c_1},\dots,^{b_n}/_{c_n}\}) \vdash P' \approx_{\mathrm{a}} Q'$. Let $\{^{B}/_{C}\} = \{^{b_1}/_{c_1},\dots,^{b_n}/_{c_n}\}$.
   By Lemma 6.4.17 we get that $h \cup \mathrm{Id}_B = \varphi(\sigma\{^{B}/_{C}\}, \rho\{^{B}/_{C}\})$, so
   $\Phi((\sigma\{^{B}/_{C}\}, \rho\{^{B}/_{C}\}), P', Q') = (h \cup \mathrm{Id}_B, P', Q') \in \Phi(\approx_{\mathrm{a}})$.
3. Assume that $P \xrightarrow{(\nu\tilde{c})\,\bar{a}\,M} P', h \vdash a \leftrightarrow b$ and $(\mathrm{fn}(P) \cup \mathrm{n}(\pi_1(h))) \cap \{\tilde{c}\} = \emptyset$.
   By Lemma 6.4.14 we get that $(\sigma,\rho) \vdash a \leftrightarrow b$. By Corollary 6.4.9 $\mathrm{n}(\pi_1(h)) = \mathrm{fn}(\sigma)$, so $\mathrm{fn}(P,\sigma) \cap \{\tilde{c}\} = \emptyset$. As $(\sigma,\rho) \vdash P \approx_{\mathrm{a}} Q$ there exists $Q', N, \tilde{d}$ such that $Q \xRightarrow{(\nu\tilde{d})\,\bar{b}\,N} Q'$, $\mathrm{fn}(Q,\rho) \cap \{\tilde{d}\} = \emptyset$ and $(\sigma\{^{M}/_{x}\}, \rho\{^{M}/_{x}\}) \vdash P' \approx_{\mathrm{a}} Q'$.
   By Corollary 6.4.9 $\mathrm{fn}(\rho) = \mathrm{n}(\pi_2(h))$, so $(\mathrm{fn}(Q) \cup \mathrm{n}(\pi_2(h))) \cap \{\tilde{d}\} = \emptyset$. According to Lemma 6.4.15 $\varphi(\sigma\{^{M}/_{x}\}, \rho\{^{N}/_{x}\}) = \mathcal{I}(h \cup \{(M,N)\})$, so $\Phi((\sigma\{^{M}/_{x}\}, \rho\{^{N}/_{x}\}), P', Q') = (\mathcal{I}(h \cup \{(M,N)\}), P', Q') \in \Phi(\approx_{\mathrm{a}})$.

□

Theorem 7.5.2 $\Phi^{-1}(\approx_{\mathrm{h}})$ *is an alley bisimulation.*

Proof. $\Phi^{-1}(\approx_{\mathrm{h}})$ is symmetric by the symmetry of $\approx_{\mathrm{h}}$. As Φ is only defined for consistent pairs of substitutions $\Phi^{-1}(\approx_{\mathrm{h}})$ is consistent. Assume that $h \vdash P \approx_{\mathrm{h}} Q$ where $h = \varphi(\sigma,\rho)$.

1. If $P \xrightarrow{\tau} P'$ then there exists Q' such that $Q \Rightarrow Q'$ and $h \vdash P' \approx_{\mathrm{h}} Q'$. Clearly $((\sigma,\rho),P',Q') \in \Phi^{-1}(h,P',Q')$.

2. Assume that $(\sigma, \rho) \vdash a \leftrightarrow b$ and that $B = \mathrm{n}(\zeta) \setminus \mathrm{dom}(\sigma)$ is such that $B \cap \mathrm{fn}(P, Q, \rho, \sigma) = \emptyset$. Furthermore, let $[\![\zeta\sigma]\!] = M, [\![\zeta\rho]\!] = N$ and assume that $P \xrightarrow{a\,M} P'$. Assume that $|B| = n$ and let $C = \{c_1, c_2, \ldots, c_n\}$ be any set of names not in $\mathrm{dom}(\sigma)$. We write $\{{}^{B}\!/_{C}\}$ for $\{{}^{b_1}\!/_{c_1}, \ldots, {}^{b_n}\!/_{c_n}\}$.
   By Lemma 6.4.17 we have that $h \cup \mathrm{Id}_B = \varphi(\sigma\{{}^{B}\!/_{C}\}, \rho\{{}^{B}\!/_{C}\})$. We also get that $h \vdash a \leftrightarrow b$ and $h \cup \mathrm{Id}_B \vdash M \leftrightarrow N$ by Lemma 6.4.13. Now, as $h \vdash P \approx_{\mathrm{h}} Q$ there is Q' such that $Q \xRightarrow{b\,N} Q'$ and $h \cup \mathrm{Id}_B \vdash P' \approx_{\mathrm{h}} Q'$. Note that $\Phi((\sigma\{{}^{B}\!/_{C}\}, \rho\{{}^{B}\!/_{C}\}), P', Q') = (h \cup \mathrm{Id}_B, P', Q')$
3. Assume that $(\sigma, \rho) \vdash a \leftrightarrow b$ and $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$ where $\mathrm{fn}(P, \sigma) \cap \{\tilde{c}\} = \emptyset$. By Lemma 6.4.13 we have that $h \vdash a \leftrightarrow b$, and by Corollary 6.4.9 $\mathrm{fn}(\sigma) = \mathrm{n}(\pi_1(h))$.

   As $h \vdash P \approx_{\mathrm{h}} Q$ there are $Q', N, \tilde{d}$ such that $Q \xRightarrow{(\nu\tilde{d})\,\overline{b}\,N} Q'$, $(\mathrm{fn}(Q) \cup \mathrm{n}(\pi_2(h))) \cap \{\tilde{d}\} = \emptyset$ and $\mathcal{I}(h \cup \{(M, N)\}) \vdash P' \approx_{\mathrm{h}} Q'$. By Corollary 6.4.9 $\mathrm{fn}(\rho) = \mathrm{n}(\pi_2(h))$, so $\mathrm{fn}(Q, \rho) \cap \{\tilde{d}\} = \emptyset$. $\mathcal{I}(h \cup \{(M, N)\}) = \varphi(\sigma\{{}^{M}\!/_{x}\}, \rho\{{}^{N}\!/_{x}\})$ by Lemma 6.4.15, so
   $\Phi((\sigma\{{}^{M}\!/_{x}\}, \rho\{{}^{N}\!/_{x}\}), P', Q') = (\mathcal{I}(h \cup \{(M, N)\}), P', Q')$

□

We now restate these results using the terminology of Definition 5.2.1.

Lemma 7.5.3 $\approx_{\mathrm{h}}$ *is fully* φ*-abstract w.r.t.* $\approx_{\mathrm{a}}$*.*

Proof. Fix $\sigma \sim \rho$. We need to check that for all processes P, Q we have that $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$ if and only if $\varphi(\sigma, \rho) \vdash P \approx_{\mathrm{h}} Q$. By Theorem 7.5.2 we have the "if" and Theorem 7.5.1 gives the "only if". □

Lemma 7.5.4 $\approx_{\mathrm{a}}$ *is fully* θ^f*-abstract w.r.t.* $\approx_{\mathrm{h}}$*.*

Proof. Fix a consistent hedge h. We need to check that for all processes P, Q we have that $h \vdash P \approx_{\mathrm{h}} Q$ if and only if $\theta^f(h) \vdash P \approx_{\mathrm{a}} Q$. By Lemma 6.4.12 we have that $\Phi(\theta^f(h), P, Q) = (h, P, Q)$. Using this, Theorem 7.5.1 gives the "if" and the "only if" follows from Theorem 7.5.2. □

By combining these results, we have

Theorem 7.5.5 $\approx_{\mathrm{a}}$ *is* $(\varphi,\ \theta^f)$*-equivalent to* $\approx_{\mathrm{h}}$*.*

By applying this theorem, we can transfer the proof techniques developed by [BDP02] to hedged bisimulation. For instance, Theorem 7.4.4 states that hedged bisimulation is sound up to weakening.

We can also use this theorem to transfer results on hedged bisimilarity to alleys. For instance, we have that $\mathcal{M}$-equivalent alleys can be substituted for each other in bisimulations.

Lemma 7.5.6 *If* $(\sigma, \rho) \gtrless (\sigma', \rho')$ *then* $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$ *if and only if* $(\sigma', \rho') \vdash P \approx_{\mathrm{a}} Q$

Proof. By Lemma 6.6.4 we have that $\varphi(\sigma, \rho) = \varphi(\sigma', \rho')$. The result follows from Lemma 7.5.3, stating that $(\sigma, \rho) \vdash P \approx_{\mathrm{a}} Q$ if and only if $\varphi(\sigma, \rho) \vdash P \approx_{\mathrm{h}} Q$. □

7.6 Negative results

Having found the full abstractions above, we can now disprove the existence of full abstractions between other pairs of bisimilarities.

Proposition 7.6.1 *In this proposition, we write "*$\approx_{\mathrm{x}}$ *is not fully abstract w.r.t.* $\approx_{\mathrm{y}}$*" for "there is no mapping* G *such that* $\approx_{\mathrm{x}}$ *is fully* G*-abstract w.r.t.* $\approx_{\mathrm{y}}$*".*

1. $\approx_{\mathrm{f}}$ *is not fully abstract w.r.t.* $\approx_{\mathrm{h}}$*,* $\approx_{\#}$*,* $\approx_{\mathrm{a}}$ *or* $\approx_{\mathrm{s}}$*.*
2. $\approx_{\#}$ *is not fully abstract w.r.t.* $\approx_{\mathrm{h}}$*,* $\approx_{\mathrm{f}}$ *or* $\approx_{\mathrm{a}}$*.*

3. $\approx_{\mathrm{h}}$ *is not fully abstract w.r.t.* $\approx_{\mathrm{f}}$*,* $\approx_{\#}$ *or* $\approx_{\mathrm{s}}$*.*
4. $\approx_{\mathrm{a}}$ *is not fully abstract w.r.t.* $\approx_{\mathrm{f}}$*,* $\approx_{\#}$ *or* $\approx_{\mathrm{s}}$*.*
5. $\approx_{\mathrm{s}}$ *is not fully abstract w.r.t.* $\approx_{\mathrm{f}}$*,* $\approx_{\mathrm{h}}$ *or* $\approx_{\mathrm{a}}$*.*

Proof. To show that there is no function $g : \mathbf{E}_{\mathrm{x}} \to \mathbf{E}_{\mathrm{y}}$ satisfying $e_{\mathrm{x}} \equiv^{\mathrm{x}}_{\mathrm{y}} g(e_{\mathrm{x}})$, we show that there is one consistent non-blind environment e_{x} that is not $\approx^{\mathrm{x}}_{\mathrm{y}}$-equivalent to any of its $\mathcal{M}$-equivalent counterparts in $\mathbf{E}_{\mathrm{y}}$. Alternatively, we may use the transitivity of full abstractness to derive a contradiction.

1. (a) The hedge $\{(a,a)\}$ is consistent but not h-blind, since $\{(a,a)\} \not\vdash \overline{a}\langle a\rangle.\,\mathbf{0} \approx_{\mathrm{h}} \mathbf{0}$. We look for a $\approx^{h}_{f}$-equivalent frame-theory pair $(\mathrm{fr},\mathrm{th})$. Proposition 5.4.3 then gives that such a frame-theory pair must be consistent and satisfy $(\mathrm{fr},\mathrm{th}) \gtrless \{(a,a)\}$.
   Since $(\mathrm{fr},\mathrm{th})$ is consistent we have by Lemma 6.2.3 that $\psi(\mathrm{fr},\mathrm{th})$ is consistent. We also have that $(\mathrm{fr},\mathrm{th}) \gtrless \psi(\mathrm{fr},\mathrm{th})$, so the transitivity of $\gtrless$ gives that $\psi(\mathrm{fr},\mathrm{th}) \gtrless \{(a,a)\}$. Since both of these hedges are consistent, Lemma 6.1.18 gives that $\psi(\mathrm{fr},\mathrm{th}) = \{(a,a)\}$. Then Lemma 6.2.2 gives us that $(\mathrm{fr},\mathrm{th}) = (\{a\},\emptyset)$, which has been shown in Section 4.2 not to relate the same processes as the hedge $\{(a,a)\}$.
   (b) By Proposition 4.1.4 there exists a consistent non-blind $(\mathrm{fr},\mathrm{th})$ such that $(\mathrm{fr},\mathrm{th}) \not\equiv^{\#}_{\mathrm{f}} (\mathrm{fr},\mathrm{th})$. We try to find a frame-theory pair $(\mathrm{fr}',\mathrm{th}')$ that is $\approx^{\#}_{\mathrm{f}}$-equivalent to $(\mathrm{fr},\mathrm{th})$.
   By Proposition 5.4.3 we have that such a frame-theory pair must be consistent and non-blind, and that $(\mathrm{fr}',\mathrm{th}') \gtrless (\mathrm{fr},\mathrm{th})$ must hold. Lemma 6.6.2 then gives that $(\mathrm{fr}',\mathrm{th}') = (\mathrm{fr},\mathrm{th})$, but $(\mathrm{fr},\mathrm{th}) \not\equiv^{\#}_{\mathrm{f}} (\mathrm{fr},\mathrm{th})$.
   (c) Assume that $\approx_{\mathrm{f}}$ is fully g-abstract w.r.t. $\approx_{\mathrm{a}}$. We have by Lemma 7.5.4 that $\approx_{\mathrm{a}}$ is fully θ^{f}-abstract w.r.t. $\approx_{\mathrm{h}}$. By transitivity this implies that $\approx_{\mathrm{f}}$ is fully $g \circ \theta^{f}$-abstract w.r.t. $\approx_{\mathrm{h}}$, which contradicts 1(a).
   (d) Assume that $\approx_{\mathrm{f}}$ is fully g-abstract w.r.t. $\approx_{\mathrm{s}}$. We have by Lemma 7.1.5 that $\approx_{\mathrm{s}}$ is fully $\theta^{f} \circ \psi$-abstract w.r.t. $\approx_{\#}$. By transitivity this implies that $\approx_{\mathrm{f}}$ is fully $g \circ \theta^{f} \circ \psi$-abstract w.r.t. $\approx_{\#}$, which contradicts 1(b).
2. As 1(a),(b),(c).
3. (a) By Proposition 4.3.5 there exists a consistent non-blind $(\mathrm{fr},\mathrm{th})$ such that $(\mathrm{fr},\mathrm{th}) \not\equiv^{\mathrm{f}}_{\mathrm{h}} \psi(\mathrm{fr},\mathrm{th})$. We try to find a hedge h that is $\approx^{\mathrm{f}}_{\mathrm{h}}$-equivalent to $(\mathrm{fr},\mathrm{th})$.
   By Proposition 5.4.3 we have that such a hedge must be consistent and non-blind, and that $h \gtrless (\mathrm{fr},\mathrm{th})$ must hold. Since $(\mathrm{fr},\mathrm{th}) \gtrless \psi(\mathrm{fr},\mathrm{th})$ we have by transitivity that $h \gtrless \psi(\mathrm{fr},\mathrm{th})$. Note that $\psi(\mathrm{fr},\mathrm{th})$ is consistent by Lemma 6.2.3. Then we can use Lemma 6.1.18 to derive that $h = \psi(\mathrm{fr},\mathrm{th})$, which is a contradiction.
   (b) As 3(a).
   (c) As 1(d).
4. (a) Assume that $\approx_{\mathrm{a}}$ is fully g-abstract w.r.t. $\approx_{\mathrm{f}}$. We have by Lemma 7.5.3 that $\approx_{\mathrm{h}}$ is fully φ-abstract w.r.t. $\approx_{\mathrm{a}}$. By transitivity this implies that $\approx_{\mathrm{h}}$ is fully $\varphi \circ g$-abstract w.r.t. $\approx_{\mathrm{f}}$, which is false by 3(a).
   (b) As 4(a).
   (c) As 1(d).
5. (a) Assume that $\approx_{\mathrm{s}}$ is fully g-abstract w.r.t. $\approx_{\mathrm{f}}$. We have by Lemma 7.1.3 that $\approx_{\#}$ is fully γ-abstract w.r.t. $\approx_{\mathrm{s}}$. By transitivity this implies that $\approx_{\#}$ is fully $\gamma \circ g$-abstract with respect to $\approx_{\mathrm{f}}$, which is false by 2(b).
   (b) As 5(a).
   (c) As 5(a).

□

We summarize the relations between the bisimilarities in Table 5. Note that there are no environment mappings from alleys or hedges to frame-theory pairs preserving both soundness and the synthesis (cf. Lemma 6.2.2).

	Alley	Hedged	Framed	Fenced	Trellis	$\approx_\top$	$\approx_\bot$
Alley		F	s,C	s,C	s,C	F	F
Hedged	F		s,C	s,C	s,C	F	F
Framed	s,c	s,c		s,C	s,C	F	F
Fenced	s,c	s,c	S,c		F	F	F
Trellis	s,c	s,c	S,c	F		F	F
$\approx_\top$	c	c	c	c	c		c
$\approx_\bot$	s	s	s	s	s	s	

In this table, an "s" means that there is a trivial function g making the bisimilarity leading the row sound with respect to the one heading the column and "c" is for trivial completeness. A "S" means that there is a "good" sound environment mapping, "C" for the existence of a "good" complete one and "F" for any full abstraction. We don't show relations that are subsumed by stronger ones.

Table 5: Relations between the bisimilarities.

8 Comparison in a Categorical Framework

Usually, bisimilarities are represented and studied as sets of process pairs; comparisons between bisimilarities are therefore based on set-theoretic comparisons. Environment-sensitive bisimilarities extend this notion by adding a third component to the process pair. Likewise, in the previous sections, we have achieved a reasonably satisfying set-theoretic understanding of the relations between the various environment-sensitive bisimilarities, which properly take into account the relations (represented by mappings) between the respective additional environment components. It turns out that we can improve on this merely set-theoretic understanding and further refine the comparison among the bisimilarities by not only studying the elements of bisimilarities, but also the (pairs of matching) transitions that connect them. The application of this technique to bisimilarities for other process calculi, e.g. the pi calculus, is a subject of possible future work.

As we have seen, definitions of bisimulations typically contain statements such as "If $e \vdash P \mathcal{R} Q$ and $P \xrightarrow{\tau} P'$ then there exists Q' such that $Q \Longrightarrow Q'$ and $e \vdash P' \mathcal{R} Q'$". This provides us with some internal structure on the set $\mathcal{R}$, namely transition pairs connecting the objects (e, P, Q) and (e, P', Q'). For bookkeeping purposes, we label such arrows with the process actions. This procedure, more formally defined below, turns each bisimulation itself into a labelled transition system, which allows us to study the internal structure of the bisimilarities. As a standard uniform framework for this kind of comparison, we use the language of category theory. We straightforwardly redefine the bisimilarities as categories, then we lift the environment mappings to functors between those categories and study the properties of these functors.

Let $\mathbf{A}$ denote the set of *actions* defined by the following grammar:

$$\mu, \gamma \quad ::= \quad \tau \quad | \quad \overline{a}\, M \quad | \quad a\, M$$

Note that new names are not explicitly mentioned on process output. However, by inspection of the derivation of $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$ we have that $\{\tilde{c}\} \subseteq \mathrm{n}(M)$. (Note that $\tilde{c} = \mathrm{n}(M) \setminus \mathrm{fn}(P)$ may be false, due to the rules Sum, Let and Guard.) Since the simulations only consider outputs where $\tilde{c}$ is fresh[4], we have that $\tilde{c}$ is simply the fresh names in $\mathrm{n}(M)$.

8.1 Redefinitions

The categorical definitions below all have a similar structure:

[4] See Section 3 for the precise meaning in each particular case.

- The objects are process pairs under some kind of environment. Processes are considered up to α-equivalence.
- The arrows are labelled, and correspond to matching transitions of the process pair. Due to the closure requirement on the composition of arrows in a category, the labels must be strings over the set of actions.
- Composition of arrows is concatenation of the labels, which clearly is associative. If $A \xrightarrow{\tilde{\mu}} B$ and $B \xrightarrow{\tilde{\gamma}} C$ then we write $A \xrightarrow{\tilde{\mu}\tilde{\gamma}} C$ for the arrow $B \xrightarrow{\tilde{\gamma}} C \circ A \xrightarrow{\tilde{\mu}} B$.
- Two arrows are considered equal if they have the same label, domain and codomain.
- The identity arrows simply correspond to not doing any transition, and are labelled with the empty string (different from τ, since identity arrows must be idempotent and $\tau\tau \neq \tau$ as strings).

Note that the arrows are only labelled with the actions of the first process in the pair. Together with the codomain, this uniquely determines the actions of the second process, since a given consistent environment can not consider a given message equivalent to two different messages. We may now proceed to the redefinitions.

Framed Bisimulation

The category $\mathfrak{F}$ has $\approx_{\mathrm{f}}$ as set of objects. We say that there is a primitive $\mathfrak{F}$-arrow from $((\mathrm{fr}, \mathrm{th}), P, Q)$ to $((\mathrm{fr}', \mathrm{th}'), P', Q')$ iff $(\mathrm{fr}', \mathrm{th}') \vdash P' \approx_{\mathrm{f}} Q'$ and one of the following conditions holds:

1. $P \xrightarrow{\tau} P'$, $Q \Rightarrow Q'$ and $(\mathrm{fr}', \mathrm{th}') = (\mathrm{fr}, \mathrm{th})$. This arrow is labelled with τ.
2. $P \xrightarrow{a\,M} P'$, $Q \xRightarrow{a\,N} Q'$, $\mathrm{th}' = \mathrm{th}$ and $\mathrm{fr}' = \mathrm{fr} \cup B$ where $a \in \mathrm{fr}$, $B \subset \mathcal{N}$ is finite, $B \cap (\mathrm{fn}(P, Q) \cup \mathrm{fr} \cup \mathrm{n}(\mathrm{th})) = \emptyset$ and $(\mathrm{fr} \cup B, \mathrm{th}) \vdash M \leftrightarrow N$. This arrow is labelled with $a\,M$.
3. $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $Q \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} Q'$, $(\mathrm{fr}, \mathrm{th}) \leq (\mathrm{fr}', \mathrm{th}')$ and $(\mathrm{fr}', \mathrm{th}') \vdash M \leftrightarrow N$ where $a \in \mathrm{fr}$,$\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{fr} \cup \mathrm{n}(\pi_1(\mathrm{th}))) = \emptyset$ and $\{\tilde{d}\} \cap (\mathrm{fn}(Q) \cup \mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th}))) = \emptyset$. This arrow is labelled with $\overline{a}\,M$.

The arrows in $\mathfrak{F}$ are the identity arrows and the transitive closure of the primitive $\mathfrak{F}$-arrows.

Fenced Bisimulation

The category $\mathfrak{F}_{\#}$ has $\approx_{\#}$ as set of objects. We say that there is a primitive $\mathfrak{F}_{\#}$-arrow from $((\mathrm{fr}, \mathrm{th}), P, Q)$ to $((\mathrm{fr}', \mathrm{th}'), P', Q')$ iff $(\mathrm{fr}', \mathrm{th}') \vdash P' \approx_{\#} Q'$ and one of the following conditions holds:

1. $P \xrightarrow{\tau} P'$, $Q \Rightarrow Q'$ and $(\mathrm{fr}', \mathrm{th}') = (\mathrm{fr}, \mathrm{th})$. This arrow is labelled with τ.
2. $P \xrightarrow{a\,M} P'$, $Q \xRightarrow{a\,N} Q'$, $\mathrm{th}' = \mathrm{th}$ and $\mathrm{fr}' = \mathrm{fr} \cup B$, where $a \in \mathrm{fr}$, $B \subset \mathcal{N}$ is finite, $B \cap (\mathrm{fn}(P, Q) \cup \mathrm{fr} \cup \mathrm{n}(\mathrm{th})) = \emptyset$ and $(\mathrm{fr} \cup B, \mathrm{th}) \vdash M \leftrightarrow N$. This arrow is labelled with $a\,M$.
3. $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $Q \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} Q'$ and $(\mathrm{fr}', \mathrm{th}') = \xi(\mathrm{fr}, \mathrm{th}, M, N)$ where $a \in \mathrm{fr}$, $\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{fr} \cup \mathrm{n}(\pi_1(\mathrm{th}))) = \emptyset$ and $\{\tilde{d}\} \cap (\mathrm{fn}(Q) \cup \mathrm{fr} \cup \mathrm{n}(\pi_2(\mathrm{th}))) = \emptyset$. This arrow is labelled with $\overline{a}\,M$.

The arrows in $\mathfrak{F}_{\#}$ are the identity arrows and the transitive closure of the primitive $\mathfrak{F}_{\#}$-arrows.

Alley Bisimulation

The category $\mathfrak{A}$ has $\approx_{\mathrm{a}}$ as set of objects. We say that there is a primitive $\mathfrak{A}$-arrow from $((\sigma, \rho), P, Q)$ to $((\sigma', \rho'), P', Q')$ iff $(\sigma', \rho') \vdash P' \approx_{\mathrm{a}} Q'$ and one of the following conditions holds:

1. $P \xrightarrow{\tau} P'$, $Q \Rightarrow Q'$, $\sigma' = \sigma$ and $\rho' = \rho$. This arrow is labelled with τ.
2. $P \xrightarrow{a\,M} P'$, $Q \xRightarrow{b\,N} Q'$, $\sigma' = \sigma\{{}^{B}/_{C}\}$ and $\rho' = \rho\{{}^{B}/_{C}\}$ where $B \cap \mathrm{fn}(P, Q, \rho, \sigma) = \emptyset$, $C \cap \mathrm{dom}(\sigma) = \emptyset$, $(\sigma, \rho) \vdash a \leftrightarrow b$ and there exists ζ such that $B = \mathrm{fn}(\zeta) \setminus \mathrm{dom}(\sigma)$, $[\![\, \zeta\sigma \,]\!] = M$ and $[\![\, \zeta\rho \,]\!] = N$.
   This arrow is labelled with $a\,M$.

3. $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $Q \overset{(\nu\tilde{d})\,\overline{b}\,N}{\Longrightarrow} Q'$, $\sigma' = \sigma\{^M/_x\}$ and $\rho' = \rho\{^N/_x\}$ where $\mathrm{fn}(P,\sigma)\cap\{\tilde{c}\} = \emptyset$, $\mathrm{fn}(Q,\rho)\cap\{\tilde{d}\} = \emptyset$ and $(\sigma,\rho) \vdash a \leftrightarrow b$. This arrow is labelled with $\overline{a}\,M$.

The arrows in $\mathfrak{A}$ are the identity arrows and the transitive closure of the primitive $\mathfrak{A}$-arrows.

Trellis Bisimulation

The category $\mathfrak{S}$ is the sub-category of $\mathfrak{A}$ obtained by restricting the set of objects to $\approx_{\mathrm{s}}$.

Hedged Bisimulation

The category $\mathfrak{H}$ has $\approx_{\mathrm{h}}$ as set of objects. We say that there is a primitive $\mathfrak{H}$-arrow from (h,P,Q) to (h',P',Q') iff $h' \vdash P' \approx_{\mathrm{h}} Q'$ and one of the following conditions holds:

1. $P \xrightarrow{\tau} P'$, $Q \Rightarrow Q'$ and $h' = h$. This arrow is labelled with τ.
2. $P \xrightarrow{a\,M} P'$, $Q \overset{b\,N}{\Longrightarrow} Q'$ and $h' = h \cup \mathrm{Id}_B$ where $h \vdash a \leftrightarrow b$, $h \cup \mathrm{Id}_B \vdash M \leftrightarrow N$, $B \subset \mathcal{N}$ is finite and $B \cap (\mathrm{fn}(P,Q) \cup \mathrm{n}(h)) = \emptyset$. This arrow is labelled with $a\,M$.
3. $P \xrightarrow{(\nu\tilde{c})\,\overline{a}\,M} P'$, $Q \overset{(\nu\tilde{d})\,\overline{b}\,N}{\Longrightarrow} Q'$ and $h' = \mathcal{I}(h \cup \{(M,N)\})$ where $h \vdash a \leftrightarrow b$, $\{\tilde{c}\} \cap (\mathrm{fn}(P) \cup \mathrm{n}(\pi_1(h))) = \emptyset$ and $\{\tilde{d}\} \cap (\mathrm{fn}(Q) \cup \mathrm{n}(\pi_2(h))) = \emptyset$. This arrow is labelled with $\overline{a}\,M$.

The arrows of $\mathfrak{H}$ are the identity arrows and the transitive closure of the primitive $\mathfrak{H}$-arrows.

8.2 Reinterpretation

We now attempt to lift our environment mappings to functors beteen the bisimilarities, and study the properties of these functors.

Framed and Fenced

We begin by comparing $\mathfrak{F}$ and $\mathfrak{F}_\#$. Regarding the objects, $\approx_\# \subsetneq \approx_{\mathrm{f}}$ by Proposition 4.1.4 and Theorem 7.3.1(1). At process output we have that $((\mathrm{fr},\mathrm{th}),P,Q) \xrightarrow{\overline{a}\,M} ((\mathrm{fr}',\mathrm{th}'),P',Q')$ in $\mathfrak{F}_\#$ only if $(\mathrm{fr}',\mathrm{th}') = \xi(\mathrm{fr},\mathrm{th},M,N)$. In $\mathfrak{F}$, we also have an arrow $\xrightarrow{\overline{a}\,M}$ from $((\mathrm{fr},\mathrm{th}),P,Q)$ to $(\xi(\mathrm{fr},\mathrm{th},M,N),P',Q')$ according to Lemma 6.3.1 and Theorem 7.3.1(1). As framed and fenced bisimulations behave identically on process input and internal calculation, there is a trivial embedding functor $\mathfrak{F}_\# \to \mathfrak{F}$. However, in $\mathfrak{F}$ we are allowed to further extend the frame-theory pair on process output, giving raise to arrows not present in $\mathfrak{F}_\#$.

Example 8.2.1 *Let $P = \overline{a}\langle a\rangle.\mathbf{0}$. Clearly $(\{a\},\emptyset) \vdash P \approx_\# P$ and $(\{a\},\emptyset) \vdash P \approx_{\mathrm{f}} P$. Except for the identity, the only arrow from $((\{a\},\emptyset),P,P)$ in $\mathfrak{F}_\#$ is $((\{a\},\emptyset),P,P) \xrightarrow{\overline{a}\,a} ((\{a\},\emptyset),\mathbf{0},\mathbf{0})$. However, in $\mathfrak{F}$ there are arrows $((\{a\},\emptyset),P,P) \xrightarrow{\overline{a}\,a} ((\mathrm{fr},\mathrm{th}),\mathbf{0},\mathbf{0})$ whenever $(\mathrm{fr},\mathrm{th})$ is consistent and $a \in \mathrm{fr}$.*

Fenced and Hedged

We define $\Psi^\# : \mathfrak{F}_\# \to \mathfrak{H}$ as $\Psi^\#((\mathrm{fr},\mathrm{th}),P,Q) = (\psi(\mathrm{fr},\mathrm{th}),P,Q)$ and $\Psi^\#(((\mathrm{fr},\mathrm{th}),P,Q) \xrightarrow{\tilde{\mu}} ((\mathrm{fr}',\mathrm{th}'),P',Q')) = (\psi(\mathrm{fr},\mathrm{th}),P,Q) \xrightarrow{\tilde{\mu}} (\psi(\mathrm{fr}',\mathrm{th}'),P',Q')$. As shown in the proof of Theorem 7.2.1, any primitive $\mathfrak{F}_\#$-arrow corresponds to a primitive $\mathfrak{H}$-arrow with the same label, so by composing these we get that $\Psi^\#$ is a functor. $\Psi^\#$ is full, since both bisimilarities perform minimal extensions, and faithful, since arrows are equal iff they have the same labels, which are preserved by $\Psi^\#$. However, there is a hedged process pair in $\mathrm{range}(\Psi^\#)$ with an $\mathfrak{H}$-arrow that leads outside $\mathrm{range}(\Psi^\#)$.

Example 8.2.2 *Let* $P = \overline{a}\langle a\rangle.(\nu l)\,\overline{a}\langle l\rangle.\,\mathbf{0} + \overline{a}\langle a\rangle.\overline{a}\langle k\rangle.\,\mathbf{0}$ *where* $k \neq a$. *We have that* $(\{a\},\emptyset) \vdash P \approx_{\#} P$*, intuitively since the simulating process can always mimic the simulated. In* $\mathfrak{H}$ *we have that* $(\{(a,a)\},P,P) \xrightarrow{\overline{a}\,a} (\{(a,a)\},(\nu l)\,\overline{a}\langle l\rangle.\,\mathbf{0},\overline{a}\langle k\rangle.\,\mathbf{0})$ *but* $(\{a\},\emptyset,(\nu l)\,\overline{a}\langle l\rangle.\,\mathbf{0},\overline{a}\langle k\rangle.\,\mathbf{0})$ *is not in* $\approx_{\#}$ *by Proposition 4.3.3 and Theorem 7.3.1.*

We also have that the obvious extention of $\Psi^{\#}$ to $\mathfrak{F}$ is not a functor, because of the multitude of possible extensions of the frame-theory pair on process output (see Example 8.2.1).

Hedged and Alley

We extend the definition of Φ to $\mathfrak{A} \to \mathfrak{H}$ by letting $\Phi(((\sigma,\rho),P,Q) \xrightarrow{\tilde{\mu}} ((\sigma',\rho'),P',Q')) := (\varphi(\sigma,\rho),P,Q) \xrightarrow{\tilde{\mu}} (\varphi(\sigma',\rho'),P',Q')$. As shown in the proof of Theorem 7.2.1, any primitive $\mathfrak{F}_{\#}$-arrow corresponds to a primitive $\mathfrak{H}$-arrow with the same label, so by composing these we get that Ψ is a functor. Ψ is faithful, since arrows are equal iff they have the same labels, which are preserved by Ψ. However, Ψ is not full, since several different non-isomorphic alley process pairs are mapped to a given hedged process pair.

We can also define $\Theta^f : \mathfrak{H} \to \mathfrak{A}$ by letting $\Theta^f(h,P,Q) := (\theta^f(h),P,Q)$ and $\Theta^f((h,P,Q) \xrightarrow{\tilde{\mu}} (h',P',Q')) := (\theta^f(h),P,Q) \xrightarrow{\tilde{\mu}} (\theta^f(h'),P',Q')$. Unfortunately Θ^f is not a functor, since on process output the environments in $\mathfrak{A}$ simply add message pairs instead of reducing them and discarding duplicates.

Example 8.2.3 *Let* $P = \overline{a}\langle \mathrm{E}_a(a)\rangle.\,\mathbf{0}$. *Clearly* $\{(a,a)\} \vdash P \approx_{\mathrm{h}} P$. *Assume that* $f(a,a) = x$ *where* f *is the function defining* θ^f. *As in the proof of Theorem 7.5.5 we have that* $(\{{}^{a}\!/\!_{x}\},\{{}^{a}\!/\!_{x}\}) \vdash P \approx_{\mathrm{a}} P$. *In* $\mathfrak{H}$ *we have that* $(\{(a,a)\},P,P) \xrightarrow{\overline{a}\ \mathrm{E}_a(a)} (\{(a,a)\},\mathbf{0},\mathbf{0})$. *A corresponding arrow in* $\mathfrak{A}$ *is* $((\{{}^{a}\!/\!_{x}\},\{{}^{a}\!/\!_{x}\}),P,P) \xrightarrow{\overline{a}\ \mathrm{E}_a(a)} \left(\left(\left\{{}^{a}\!/\!_{x},\ {}^{\mathrm{E}_a(a)}\!/\!_{y}\right\},\left\{{}^{a}\!/\!_{x},\ {}^{\mathrm{E}_a(a)}\!/\!_{y}\right\}\right),\mathbf{0},\mathbf{0}\right)$. *However, we have that* $\Theta^f(\{(a,a)\},\mathbf{0},\mathbf{0}) = ((\{{}^{a}\!/\!_{x}\},\{{}^{a}\!/\!_{x}\}),\mathbf{0},\mathbf{0}) \neq \left(\left(\left\{{}^{a}\!/\!_{x},\ {}^{\mathrm{E}_a(a)}\!/\!_{y}\right\},\left\{{}^{a}\!/\!_{x},\ {}^{\mathrm{E}_a(a)}\!/\!_{y}\right\}\right),\mathbf{0},\mathbf{0}\right)$.

One way to fix this problem is the application of up-to techniques.

8.3 Up-to Techniques

The application of sound up-to techniques corresponds to adding up-to actions to $\mathbf{A}$. All combinations of the following up-to techniques are proved [BDP02] to be sound for $\approx_{\mathrm{a}}$:

- Up to structural congruence, which corresponds to adding s to the set of actions and adding all arrows $((\sigma,\rho),P,Q) \xrightarrow{s} ((\sigma,\rho),P',Q')$ where $P \equiv P'$ and $Q \equiv Q'$.
- Up to weakening, which corresponds to adding w to the set of actions and adding all arrows $((\sigma,\rho),P,Q) \xrightarrow{w} ((\sigma\{{}^{M}\!/\!_{x}\},\rho\{{}^{N}\!/\!_{x}\}),P,Q)$.
- Up to contraction, which corresponds to adding c to the set of actions and adding all arrows $((\sigma\{{}^{M}\!/\!_{x}\},\rho\{{}^{N}\!/\!_{x}\}),P,Q) \xrightarrow{c} ((\sigma,\rho),P,Q)$ where $(\sigma,\rho) \vdash M \leftrightarrow N$.
- Up to restriction, which corresponds to adding r to the set of actions and adding all arrows $((\sigma,\rho),(\nu\tilde{m})\,P,(\nu\tilde{n})\,Q) \xrightarrow{r} ((\sigma,\rho),P,Q)$ where $\{\tilde{m}\} \cap \mathrm{fn}(\sigma) = \emptyset$ and $\{\tilde{n}\} \cap \mathrm{fn}(\rho) = \emptyset$. When using this up-to technique, it is preferable not to record the creation of fresh names on process output.
- Up to parallel composition, which corresponds to adding p to the set of actions and adding all arrows $((\sigma,\rho),P \mid F\sigma,Q \mid F\rho) \xrightarrow{p} ((\sigma,\rho),P,Q)$ where $\mathrm{fn}(F) \subseteq \mathrm{dom}(\sigma)$.

The environment equivalences defined in Section 5 give us two additional up-to techniques:

- Up to $\approx^{a}_{\mathrm{a}}$-equivalence, which corresponds to adding an "up to $\approx^{a}_{\mathrm{a}}$-equivalence" action e to the set of actions and adding all arrows $((\sigma,\rho),P,Q) \xrightarrow{e} ((\sigma',\rho'),P,Q)$ where $(\sigma,\rho) \equiv^{a}_{a} (\sigma',\rho')$.

- Up to $\mathcal{M}$-equivalence, which means adding an "up to $\mathcal{M}$-equivalence" action m to the set of actions and adding all arrows $((\sigma, \rho), P, Q) \xrightarrow{m} ((\sigma', \rho'), P, Q)$ where $(\sigma, \rho) \gtrless (\sigma', \rho')$. Note that this up-to technique is a special case of "up to weakening and contraction", so it is sound for alley bisimulation.

After adding arrows, we must ensure that all compositions are defined. A priori, since we distinguish arrows having different labels we are sensitive to exactly where and which up-to techniques were used in the composition of an arrow, which is rarely desirable. However, we can make arrows that only differ in their use of up-to techniques equal by taking the quotient with the following equivalence:

Definition 8.3.1 *The arrows $A \xrightarrow{\tilde{\gamma}} B$ and $A \xrightarrow{\tilde{\mu}} B$ are up-to equivalent, written $\xrightarrow{\tilde{\gamma}} \equiv_u \xrightarrow{\tilde{\mu}}$ iff $\tilde{\gamma} \backslash U = \tilde{\mu} \backslash U$, i.e., $\tilde{\gamma}$ and $\tilde{\mu}$ are equal after removal of names of "up-to actions".*

In the resulting category, all arrows only labelled with reversible up-to actions are isomorphisms.

Framed and Hedged

Apart from the environments the main difference between framed and hedged bisimulations is that we in framed bisimulation may extend the process pair more than strictly necessary on process output. Since we already have a mapping on the objects of the categories, to get an embedding it should intuitively suffice to permit arbitrary extensions also on the hedged side. This corresponds to taking hedged bisimilarity up to weakening, a proof technique which has been proven sound in Theorem 7.4.4.

We let $\mathfrak{H}^w$ be $\mathfrak{H}$ up to weakening and $\equiv_u$, i.e., the category obtained from $\mathfrak{H}$ by first adding all arrows of the type $(h, P, Q) \xrightarrow{w} (h', P, Q)$ where $h \leq h'$ and the codomain actually is an object in the category, then closing the set of arrows under composition and finally taking the quotient with $\equiv_u$ as defined in Definition 8.3.1. The effect of this is that we may add information to the hedge at any time — not only on process output! However, as we are only looking for an embedding of $\mathfrak{F}$ in $\mathfrak{H}^w$, the extra arrows resulting from our preference to use a standard up-to tecnique will not cause any problems.

We then define $\Psi^w : \mathfrak{F} \to \mathfrak{H}^w$ as $\Psi^w((\mathrm{fr}, \mathrm{th}), P, Q) = (\psi(\mathrm{fr}, \mathrm{th}), P, Q)$ on objects. On primitive $\mathfrak{F}$-arrows, Ψ^w acts in the following way.

1. We define $\Psi^w(((\mathrm{fr}, \mathrm{th}), P, Q) \xrightarrow{\tau} ((\mathrm{fr}, \mathrm{th}), P', Q'))$
as the arrow $(\psi(\mathrm{fr}, \mathrm{th}), P, Q) \xrightarrow{\tau} (\psi(\mathrm{fr}, \mathrm{th}), P', Q')$.
2. We define $\Psi^w(((\mathrm{fr}, \mathrm{th}), P, Q) \xrightarrow{a\,M} ((\mathrm{fr}', \mathrm{th}'), P', Q'))$
as the arrow $(\psi(\mathrm{fr}, \mathrm{th}), P, Q) \xrightarrow{a\,M} (\psi(\mathrm{fr}', \mathrm{th}'), P', Q'))$.
3. If $((\mathrm{fr}, \mathrm{th}), P, Q) \xrightarrow{\overline{a}\,M} ((\mathrm{fr}', \mathrm{th}'), P', Q')$ then by definition there is N such that $(\mathrm{fr}', \mathrm{th}') \vdash M \leftrightarrow N$. In $\mathfrak{H}$, there is an arrow $(\psi(\mathrm{fr}, \mathrm{th}), P, Q) \xrightarrow{\overline{a}\,M} (\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}), P', Q')$, so we have to bridge the gap between the codomains of these arrows.
Fortunately, $(\mathrm{fr}, \mathrm{th}) \leq (\mathrm{fr}', \mathrm{th}')$ by the definition of framed bisimulation. By Lemma 7.4.5 we then get that $\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}) \leq \psi((\mathrm{fr}', \mathrm{th}'))$, so there is a weakening arrow $(\mathcal{I}(\psi(\mathrm{fr}, \mathrm{th}) \cup \{(M, N)\}), P', Q') \xrightarrow{w} (\psi((\mathrm{fr}', \mathrm{th}')), P', Q')$ bridging the gap. We then define $\Psi^w(((\mathrm{fr}, \mathrm{th}), P, Q) \xrightarrow{\overline{a}\,M})((\mathrm{fr}', \mathrm{th}'), P', Q') := \xrightarrow{w} \circ \xrightarrow{\overline{a}\,M}$, where the domains and codomains of the arrows on the right are as seen above.

Inductively, the action of Ψ^w on composite $\mathfrak{F}$-arrows is just the composition of the application of Ψ^w on the primitive decomposition.

Ψ^w is a faithful functor, since it is injective on both objects and labels.

8.4 Θ^f Again!

In the cases of framed, fenced and hedged bisimulation we have that two environments are $\mathcal{M}$-equal iff they are equal, so all m-arrows will disappear under $\equiv_u$. However, for substitutions we will see that "up to $\mathcal{M}$-equivalence" buys us something.

We let $\mathfrak{A}^m$ be $\mathfrak{A}$ up to $\mathcal{M}$-equivalence and $\equiv_u$, i.e., the category obtained from $\mathfrak{A}$ by first adding all arrows of the type $((\sigma,\rho),P,Q) \xrightarrow{m} ((\sigma',\rho'),P,Q)$ where $(\sigma,\rho) \gtrless (\sigma',\rho')$, then closing the set of arrows under composition and finally taking the quotient with $\equiv_u$ as defined in Definition 8.3.1. The effect of this is to make $\mathcal{M}$–equivalent environments isomorphic.

We define $\Phi : \mathfrak{A}^m \to \mathfrak{H}$ as $\Phi((\sigma,\rho),P,Q) := (\varphi(\sigma,\rho),P,Q)$ and
$\Phi(((\sigma,\rho),P,Q) \xrightarrow{\tilde{\mu}} ((\sigma',\rho'),P',Q')) := (\varphi(\sigma,\rho),P,Q) \xrightarrow{\tilde{\mu}'} (\varphi(\sigma',\rho'),P',Q')$. where $\tilde{\mu}' = \tilde{\mu} \setminus m$. The removal of the m–actions is valid since $\varphi(\sigma,\rho) = \varphi(\sigma',\rho')$ if and only if $(\sigma,\rho) \gtrless (\sigma',\rho')$ according to Lemma 6.6.4. As in the proof of Theorem 7.5.1 we have that Φ is a functor.

We also define $\Theta^f : \mathfrak{H} \to \mathfrak{A}^m$ by letting $\Theta^f(h,P,Q) := (\theta^f(h),P,Q)$. The action of Θ^f on the primitive $\mathfrak{H}$-arrows is as follows:

1. If $(h,P,Q) \xrightarrow{\tau} (h',P',Q')$ then we define $\Theta^f(\xrightarrow{\tau}) := \xrightarrow{\tau}$.
2. If $(h,P,Q) \xrightarrow{a\,M} (h',P',Q')$ we let $B = \pi_1(h' \setminus h)$. We have that $\Theta^f(h,P,Q) \xrightarrow{a\,M} ((h_1^f\{{}^{B}/_{C}\}, h_2^f\{{}^{B}/_{C}\}),P',Q')$ by the proof of Theorem 7.5.1. By Lemma 6.4.17 there is an arrow $((h_1^f\{{}^{B}/_{C}\}, h_2^f\{{}^{B}/_{C}\}),P',Q') \xrightarrow{m} (\theta^f(h'),P',Q')$. We then define $\Theta^f(\xrightarrow{a\,M}) := \xrightarrow{m} \circ \xrightarrow{a\,M}$.
3. If $(h,P,Q) \xrightarrow{\overline{a}\,M} (h',P',Q')$ then we let b and N be the messages corresponding to a respective M according to h'. As in the proof of Theorem 7.5.2 we have that $\Theta^f(h,P,Q) \xrightarrow{\overline{a}\,M} ((h_1^f\{{}^{M}/_{x}\}, h_2^f\{{}^{N}/_{x}\}),P',Q')$. By Lemma 6.4.15 we have that $\varphi(h_1^f\{{}^{M}/_{x}\}, h_2^f\{{}^{N}/_{x}\}) = h'$, so there is an arrow $((h_1^f\{{}^{M}/_{x}\}, h_2^f\{{}^{N}/_{x}\}),P',Q') \xrightarrow{m} (\theta^f(h'),P',Q')$. We then define $\Theta^f(\xrightarrow{\overline{a}\,M}) := \xrightarrow{m} \circ \xrightarrow{\overline{a}\,M}$.

By the above definition, it is clear that Θ^f is a functor.

Theorem 8.4.1 *$\mathfrak{A}^m$ and $\mathfrak{H}$ are equivalent. More precisely, $\Phi \circ \Theta^f = \mathrm{Id}_{\mathfrak{H}}$ and $\Theta^f \circ \Phi$ is isomorphic to $\mathrm{Id}_{\mathfrak{A}^m}$.*

Proof. By Lemma 6.4.12 we have that $h = \varphi(h_1^f, h_2^f)$ whenever h is consistent, so $(\Phi \circ \Theta^f)(h,P,Q) = (h,P,Q)$. Since Φ throws away all m-arrows introduced by Θ^f we have that $\Phi \circ \Theta^f = \mathrm{Id}_{\mathfrak{H}}$.
To show that $\Theta^f \circ \Phi$ is isomorphic to $\mathrm{Id}_{\mathfrak{A}^m}$ we need to find $\mathfrak{A}^m$-isomorphisms to complete the "naturality square" below. We want to use the m-arrows $((\sigma,\rho),P,Q) \xrightarrow{m} ((\theta_1^f(\varphi(\sigma,\rho)), \theta_2^f(\varphi(\sigma,\rho))),P,Q)$. There are such arrows according to Lemma 6.4.12 and they are isomorphisms since we identify $\equiv_u$-equivalent arrows and $\mathcal{M}$-equivalence is symmetric. To exhibit the functor isomorphism we need to show that the following diagram commutes whenever $((\sigma,\rho),P,Q) \xrightarrow{\tilde{\mu}} ((\sigma',\rho'),P',Q')$.

$$\begin{tikzcd}
((\sigma,\rho),P,Q) \arrow[r, "m"] \arrow[d, "\tilde{\mu}"'] & \Theta^f \circ \Phi((\sigma,\rho),P,Q) \arrow[d, "\Theta^f \circ \Phi(\tilde{\mu})"] \\
((\sigma',\rho'),P',Q') \arrow[r, "m"] & \Theta^f \circ \Phi((\sigma',\rho'),P',Q')
\end{tikzcd}$$

Consider the arrows $((\sigma,\rho),P,Q) \xrightarrow{\tilde{\mu}\,m} \Theta^f \circ \Phi((\sigma',\rho'),P',Q')$ and
$((\sigma,\rho),P,Q) \xrightarrow{m\,\Theta^f \circ \Phi(\tilde{\mu})} \Theta^f \circ \Phi((\sigma',\rho'),P',Q')$. As Θ^f and Φ only change labels by inserting and removing m-actions we have that $\xrightarrow{\tilde{\mu}\,m} \equiv_u \xrightarrow{m\,\Theta^f \circ \Phi(\tilde{\mu})}$. As the arrows in $\mathfrak{A}^m$ are equivalence classes with respect to $\equiv_u$ the diagram is commutative. □

In other words, $\mathfrak{H}$ is equivalent to "$\mathfrak{A}$ up to $\mathcal{M}$-equivalence and $\equiv_u$", where the isomorphism on the alley side is the normalization of environments given by $(\sigma,\rho) \mapsto (\theta_1^f(\varphi(\sigma,\rho)), \theta_2^f(\varphi(\sigma,\rho)))$.

We can characterize the relationship between $\mathfrak{F}_\#$ (fenced) and $\mathfrak{S}$ (trellis) in the very same way, namely that $\mathfrak{F}_\#$ is equivalent to "$\mathfrak{S}$ up to $\mathcal{M}$-equivalence and $\equiv_u$" ($\mathfrak{S}^m$), where the isomorphism on the alley side is the same normalization as above.

8.5 Summary

The relations between the different categories are graphically presented in Figure 3. With these results, we have related the bisimulations not only based on the distinguishing power of the environments, but also based on their internal structure.

$$\begin{tikzcd}
 & & & \mathfrak{S} \arrow[dl, hook] \arrow[dd, hook] \\
 & \mathfrak{F}_\# \arrow[dl, hook] \arrow[dd, hook, "\Psi"'] \arrow[r, leftrightarrow, "\Theta^f \circ \Psi", "\Gamma"'] & \mathfrak{S}^m \arrow[dd, hook] & \\
\mathfrak{F} \arrow[dd, hook, "\Psi^w"'] & & & \mathfrak{A} \arrow[dl, hook] \\
 & \mathfrak{H} \arrow[dl, hook] \arrow[r, leftrightarrow, "\Theta^f", "\Phi"'] & \mathfrak{A}^m & \\
\mathfrak{H}^w & & &
\end{tikzcd}$$

Figure 3: Categorical Relations

9 Conclusions

As an interpretation of the results of Section 4, we may underline two different deficiencies in the original definition of framed bisimulation. In a sense, it is at the same time too weak and too strong with respect to barbed equivalence, for orthogonal reasons.

1. The definition is too weak in the sense that its authors did not impose a minimality requirement on the environment and argue that this "results in simpler definitions, and does not compromise soundness (w.r.t. testing equivalence)". However, when adding the minimality requirement, as done in fenced bisimulation, the relation becomes strictly stronger than barbed equivalence. As seen in the example in Section 4.1, it is not obvious how to choose the non-minimal extention on process output in order to get a bisimilar framed process pair. Thus, for example the purpose of mechanization, we regard fenced bisimularity as better suited than framed, but both bisimilarities suffer from the second problem of being too strong.
2. The definition is too rigid, because it requires the syntactic coincidence of names received by the environment from the two processes under observation: whereas framed bisimulation requires identity, hedged bisimulation allows the environment to simply record that the names respectively received from the processes in a bisimulation step correspond (c.f. Section 4.3). This is the main reason why fenced bisimilarity does not coincide with barbed equivalence, while hedged bisimilarity does.

In Section 5, we developed a uniform framework for comparing environment-sensitive bisimilarities based on their environments, because the standard merely set-theoretic framework turned out not to be sufficiently precise. We then used this new framework to reformulate and refine the results of the two previous comparisions of environment-sensitive bisimilarities in [EHHO99] and

[FHJ01]. We found that [FHJ01] showed only the full abstraction of $\approx_{\#}$ with respect to $\approx_{\mathrm{s}}$, but not the other direction. Once this was clear, the missing full abstraction was easy to construct (c.f. Lemma 7.1.5).

We exercise the comparison framework in Section 7. In particular, we find that there exists a "framed-style" environment-sensitive bisimilarity, namely hedged, that is equivalent to barbed equivalence. We then use the distinguishing examples of Section 4 to show that we have fully characterized all full abstractions and equivalences between the bisimilarities.

In Section 8 we propose a (to our knowledge) novel method for comparing environment-sensitive bisimilarities as categories. This technique allows us to study the internal structures of the bisimilarities. In particular, it allows us to highlight the fact that the non-minimality of framed bisimilarity distinguishes it conceptually from the other bisimilarities. After redefining some up-to techniques in this setting, we are at least able to embed framed bisimilarity into "hedged bisimilarity up to weakening". We also show that hedged bisimilarity is categorically equivalent to "alley bisimilarity up to $\mathcal{M}$-equivalence", thus refining the merely set-theoretic bisimilarity equivalence that we proved in Section 7.

Further work

The spi-calculus defined in Section 2 has a very simple message syntax compared to the original spi-calculus [AG99]. However, as we see in Appendix A, our results are easy to adopt to a calculus with pairing, and there is no reason to expect that compound keys, public-key cryptography or even the addition of arbitrary function symbols as done by [Cor02] would be more difficult to treat. We chose the more restricted setting of this paper only in order to keep the definitions of the various bisimilarities as simple as possible.

From a rather subjective point of view, when working with concrete examples we find hedged bisimulation easier to work with than alley bisimulation, because the knowledge available to the environment in each reachable configuration is arguably easier to understand, and the verification of consistency is more straightforward.

The mechanization of equivalence-checking in the spi-calculus represents a major goal for our current and future work. Usually, the main problem is due to the unavoidably infinite number of transitions on process input as described in the standard operational semantics. To remedy this, similar to approaches in the pi calculus, we are currently studying symbolic semantics for the spi calculus as the basis of symbolic notions of bisimulation. Also to this aim, we use (refinements of) hedged bisimulation rather than alley bisimulation, because the involved data structures are easier to deal with. Furthermore, our prototype implementation of an equivalence-checker for hedged bisimilarity also profits from the minimality requirements of hedged bisimilarity.

A Pairing

In this appendix we add pairing to the calculus and revisit some definitions and results on hedges.

A.1 Definitions

The syntax of messages and expressions are expanded with pairing, i.e.

$$\begin{array}{lllllllllllll}
M, N & ::= & a & \mid & \mathrm{E}_k(M) & \mid & (M \,.\, N) \\[1em]
\delta & ::= & a & \mid & \mathrm{E}_\delta(\delta) & \mid & (\delta \,.\, \delta) \\[1em]
\eta, \zeta & ::= & a & \mid & \mathrm{E}_\zeta(\eta) & \mid & \mathrm{D}_\zeta(\eta) & \mid & (\eta \,.\, \zeta) & \mid & \pi_1(\eta) & \mid & \pi_2(\eta)
\end{array}$$

The evaluation function for expressions is also expanded with rules for the new constructions.

$$\llbracket (\eta \,.\, \zeta) \rrbracket \quad = \quad \begin{cases} (M \,.\, N) & \text{if } \llbracket \eta \rrbracket = M \in \mathcal{M} \text{ and } \llbracket \zeta \rrbracket = N \in \mathcal{M} \\ \bot & \text{otherwise} \end{cases}$$

$$\llbracket \pi_1(\eta) \rrbracket \quad = \quad \begin{cases} M & \text{if } \llbracket \eta \rrbracket = (M \,.\, N) \in \mathcal{M} \\ \bot & \text{otherwise} \end{cases}$$

$$\llbracket \pi_2(\eta) \rrbracket \quad = \quad \begin{cases} N & \text{if } \llbracket \eta \rrbracket = (M \,.\, N) \in \mathcal{M} \\ \bot & \text{otherwise} \end{cases}$$

Definition A.1.1 *We say that a hedge h is pair-free if whenever $(M, N) \in h$ we have that neither M nor N is a message pair.*

The concept of consistent hedges, defined in Definition 3.3.2, is adapted as follows:

Definition A.1.2 *A hedge h is consistent iff h is pair-free and the conditions of Definition 3.3.2 hold.*

The synthesis of frame-theory pairs and hedges are expanded with the following rule:

$$(\textsc{Syn-join})\ \frac{(M_1, N_1) \in \mathcal{S}(\mathrm{fr}, \mathrm{th}) \qquad (M_2, N_2) \in \mathcal{S}(\mathrm{fr}, \mathrm{th})}{((M_1 \,.\, M_2), (N_1 \,.\, N_2)) \in \mathcal{S}(\mathrm{fr}, \mathrm{th})}$$

Symmetrically, the analysis of a hedge additionally satisfies the rules

$$(\textsc{Ana-split}1)\ \frac{((M_1 \,.\, M_2), (N_1 \,.\, N_2)) \in \mathcal{A}(h)}{(M_1, N_1) \in \mathcal{A}(h)}$$

$$(\textsc{Ana-split}2)\ \frac{((M_1 \,.\, M_2), (N_1 \,.\, N_2)) \in \mathcal{A}(h)}{(M_2, N_2) \in \mathcal{A}(h)}$$

The irreducibles if a hedge is redefined as

$$\begin{aligned} \mathcal{I}(h) := \mathcal{A}(h) \setminus (&\{(\mathrm{E}_a(M), \mathrm{E}_b(N)) | (a, b) \in \mathcal{A}(h), (\mathrm{E}_a(M), \mathrm{E}_b(N)) \in \mathcal{A}(h)\} \\ &\cup \{((M_1 \,.\, M_2), (N_1 \,.\, N_2)) | ((M_1 \,.\, M_2), (N_1 \,.\, N_2)) \in \mathcal{A}(h)\}) \end{aligned}$$

A.2 Applications to Hedges

We have a corresponding change to Lemma 6.1.7:

Lemma A.2.1 *A hedge h is irreducible iff the following conditions hold:*

1. *If $(\mathrm{E}_a(M), \mathrm{E}_b(N)) \in h$ then $(a, b) \notin h$.*
2. *If $(M, N) \in h$ then at most one of M and N is a pair.*

Proof. If this holds then we can apply neither Ana-split1, Ana-split2 nor Ana-dec to any pair in h, so $\mathcal{A}(h) = h$. By the definition of $\mathcal{I}(h)$ we then have that $\mathcal{I}(h) = h$.

If h is irreducible then the disjointness holds by the definition of $\mathcal{I}(h)$, using that $\mathcal{A}(h) \supseteq \mathcal{I}(h)$. □

Note that an irreducible hedge is not always pair-free. As an example we have that $\{(a, (b \,.\, c))\}$ is irreducible but not pair-free.

To show Lemma 6.1.4 and Lemma 6.1.11 we need the following lemma, which is a companion to Lemma 6.1.1:

Lemma A.2.2 *If $h \vdash (M_1 \,.\, M_2) \leftrightarrow \mathrm{E}_a(N)$ then $((M_1 \,.\, M_2), \mathrm{E}_a(N)) \in h$. If $h \vdash \mathrm{E}_a(M) \leftrightarrow (N_1 \,.\, N_2)$ then $(\mathrm{E}_a(M), (N_1 \,.\, N_2)) \in h$.*

Proof. Clearly, neither SYN-ENC nor SYN-JOIN can derive the correspondence. □

In order to prove Lemma 6.1.19 we need to use the following result to show that $\mathcal{I}(g)$ is pair-free.

Lemma A.2.3 *If g is irreducible, h is pair-free and $g \leq h$ then g is pair-free.*

Proof. Assume that $(M, N) \in g$ and note that $h \vdash M \leftrightarrow N$. Assume that M is a pair. As h is pair-free $(M, N) \notin h$, so we must have used SYN-JOIN to derive that $h \vdash M \leftrightarrow N$. Then N is a pair. According to Lemma A.2.1 this implies that g is not irreducible, which is a contradiction. A symmetric argument holds if N is a pair. □

Otherwise, we only need to add cases for pairing, which are generally simpler than the encryption/decryption cases, to derive the results on hedges in Section 6.

For framed bisimulation,we require a consistent frame-theory pair to have a pair-free theory in addition to the requirements of Definition 3.1.2. Then the results and proofs relating frames and hedges in Section 6 hold without modification.

The definitions of framed and hedged bisimulation are unchanged. For fenced bisimulation, we need to add pair-splitting operations to ξ. The problem with defining alley bisimulation in the presence of pairing isa that a message may have several different cores. This can be treated by considering the locations of the cores (see [BDP02] for details).

We have not proved the relations between the bisimilarities in the presence of pairing, but strongly believe that they should hold. For the proofs, the only major difference should be in Lemma 6.4.10, where we need to take the new consistency requirements for alleys into account in order to show the alternative definition of φ.

A.3 Framed vs. Hedged — with pairing

As an example, we study an adaptation of the process pair used in by Abadi and Gordon in [AG98] to conjecture that framed bisimilarity is not complete wrt barbed equivalence. We define

$$\begin{aligned} P &= (\nu k, m, n)\, \overline{a}\langle \mathrm{E}_k(m \,.\, n)\rangle.(\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}) \\ Q &= (\nu k, n)\, \overline{a}\langle \mathrm{E}_k(n)\rangle.\overline{a}\langle n\rangle.\,\mathbf{0} \end{aligned}$$

As above, we wish to show that $\{(a, a)\} \vdash P \approx_{\mathrm{h}} Q$ and that $(\{a\}, \emptyset) \not\vdash P \approx_{\mathrm{f}} Q$.

Proposition A.3.1 *A hedged bisimulation $\mathcal{R}$ such that $\{(a, a)\} \vdash P \mathrel{\mathcal{R}} Q$ is defined by*

$$\begin{aligned} \mathcal{R} = &\{(\{(a, a)\}, P, Q)\} \\ &\cup \{(h(k, m, n), (\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}), \overline{a}\langle m\rangle.\,\mathbf{0}) | k, m, n \in \mathcal{N} \setminus \{a\}\} \\ &\cup \{(h(k, m, n) \cup \{(m, n)\}, \mathbf{0}, \mathbf{0}) | k, m, n \in \mathcal{N} \setminus \{a\}\} \\ &\cup \{(h(k, m, n) \cup \{(m, m)\}, \mathbf{0}, \mathbf{0}) | k, m, n \in \mathcal{N} \setminus \{a\}\} \end{aligned}$$

where $h(k, m, n) = \{(a, a), (\mathrm{E}_k(m \,.\, n), \mathrm{E}_k(n))\}$ and k, m, n are pairwise different wherever they occur.

Proof. Note that $\mathcal{R}$ is "trivially" consistent as we never receive a nor the outermost keys of encrypted messages.

- As $P \xrightarrow{(\nu k,m,n)\, \overline{a}\ \mathrm{E}_k(m\,.\,n)} (\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0})$ whenever k, m, n and a are pairwise different, we need to check that for all k, m, n there exist $Q', N, \tilde{d}$ such that $Q \xRightarrow{(\nu \tilde{d})\, \overline{a}\, N} Q', a \notin \{\tilde{d}\}$ and
$\mathcal{I}(\{(a, a), (\mathrm{E}_k(m \,.\, n), N)\}) \vdash (\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}) \mathrel{\mathcal{R}} Q'$.
We may choose $\tilde{d} = (k, n)$, $N = \mathrm{E}_k(n)$ and $Q' = \overline{a}\langle n\rangle.\,\mathbf{0}$. As $h(k, m, n)$ is consistent $\mathcal{I}(h(k, m, n)) = h(k, m, n)$ by Lemma 6.1.16.

- The output transitions of Q can be handled in the same way.
- As $(\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}) \xrightarrow{\overline{a}\,m} \mathbf{0}$ we must choose a matching transition of $\overline{a}\langle n\rangle.\,\mathbf{0}$. The only transition is valid.
- As $(\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}) \xrightarrow{\overline{a}\,n} \mathbf{0}$ we must choose a matching transition of $\overline{a}\langle n\rangle.\,\mathbf{0}$. The only transition is valid.
- As $\overline{a}\langle n\rangle.\,\mathbf{0} \xrightarrow{\overline{a}\,n} \mathbf{0}$ we must choose a matching transition of the process $(\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0})$. Both transitions are valid.

□

Proposition A.3.2 *Let P, Q as in Proposition A.3.1. Then there is no frame-theory pair* $(\mathrm{fr}, \mathrm{th})$ *such that $a \in \mathrm{fr}$ and* $(\mathrm{fr}, \mathrm{th}) \vdash P \approx_{\mathrm{f}} Q$.

Proof. Assume that there exists a framed bisimulation $\mathcal{S}$ such that $(\{a\}, \emptyset) \vdash P \;\mathcal{S}\; Q$.

1. As a is in the frame and $P \xrightarrow{(\nu k,m,n)\,\overline{a}\ \mathrm{E}_k(m\,.\,n)} (\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0})$ whenever k, m, n and a are pairwise different, we need to check that for all such k, m, n there exist $Q', N, \tilde{d}, \mathrm{fr}, \mathrm{th}$ such that $Q \xRightarrow{(\nu\tilde{d})\,\overline{a}\,N} Q', a \notin \{\tilde{d}\}$, $a \in \mathrm{fr}$ and $(\mathrm{fr}, \mathrm{th}) \vdash (\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}) \;\mathcal{S}\; Q'$. Any transition of Q is of the form $Q \xRightarrow{(\nu k',n')\,\overline{a}\ \mathrm{E}_{k'}(n')} \overline{a}\langle n'\rangle.\,\mathbf{0}$.
2. Since $(\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}) \xrightarrow{\overline{a}\,m} \mathbf{0}$ we need to check that $\overline{a}\langle n'\rangle.\,\mathbf{0}$ can do a matching transition. As the only possibility is $\overline{a}\langle n'\rangle.\,\mathbf{0} \xrightarrow{\overline{a}\,n'} \mathbf{0}$ there exists a consistent frame-theory pair $(\mathrm{fr}_1, \mathrm{th}_1)$ such that $(\mathrm{fr}_1, \mathrm{th}_1) \vdash m \leftrightarrow n'$, which can only be the case if $m = n' \in \mathrm{fr}_1$.
3. Since $(\overline{a}\langle m\rangle.\,\mathbf{0} + \overline{a}\langle n\rangle.\,\mathbf{0}) \xrightarrow{\overline{a}\,n} \mathbf{0}$ we need to check that $\overline{a}\langle n'\rangle.\,\mathbf{0}$ can do a matching transition. As the only possibility is $\overline{a}\langle n'\rangle.\,\mathbf{0} \xrightarrow{\overline{a}\,n'} \mathbf{0}$ there exists a consistent frame-theory pair $\mathrm{fr}_2, \mathrm{th}_2$ such that $(\mathrm{fr}_2, \mathrm{th}_2) \vdash n \leftrightarrow n'$, which can only be the case if $n = n' \in \mathrm{fr}_2$.
4. We thus have that $m = n$, which is a contradiction.

□

References

[AF01] Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *Proceedings of POPL '01*. ACM, January 2001.

[AG98] Martín Abadi and Andrew D. Gordon. A bisimulation method for cryptographic protocols. *Nordic Journal of Computing*, 5(4):267–303, Winter 1998.

[AG99] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The Spi calculus. *Journal of Information and Computation*, 148(1):1–70, 1999.

[BDP99] Michele Boreale, Rocco De Nicola, and Rosario Pugliese. Proof techniques for cryptographic processes. In *Proceedings of LICS '99*, pages 157–166. IEEE, Computer Society Press, July 1999.

[BDP02] Michele Boreale, Rocco De Nicola, and Rosario Pugliese. Proof techniques for cryptographic processes. *SIAM Journal on Computing*, 31(3):947–986, 2002.

[Bor01] Johannes Borgström. *On Bisimulations for the Spi-Calculus.* Term project, EPFL, Switzerland, July 2001.

[Cor02] Veronique Cortier. Observational equivalence and trace equivalence in an extension of spi-calculus. application to cryptographic protocols analysis. extended version. Technical Report LSV-02-3, Lab. Specification and Verification, ENS de Cachan, March 2002.

[DY83] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, March 1983.

[EHHO99] Anders Strandløv Elkjær, Micheal Höhle, Hans Hüttel, and Kasper Overgård. Towards automatic bisimilarity checking in the spi calculus. In C. S. Calude and M. J. Dinneen, editors, *Combinatorics, Computation & Logic*, volume 21(3) of *Australian Computer Science Communications*, pages 175–189. Springer-Verlag Singapore Pte. Ltd., January 1999.

[FHJ01] Ulrik Frendrup, Hans Hüttel, and Jesper Nyholm Jensen. Two notions of environment sensitive bisimilarity for spi-calculus processes. 2001.

[Hüt02] Hans Hüttel. Deciding framed bisimilarity. In *Pre-proceedings of Infinity'02*, pages 1–20, june 2002.

[Mil99] Robin Milner. *Communicating and Mobile Systems: the π-Calculus.* Cambridge University Press, May 1999.

[MPW93] Robin Milner, Joachim Parrow, and David Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114:149–171, 1993.

[MS92] Robin Milner and Davide Sangiorgi. Barbed bisimulation. In W. Kuich, editor, *Proceedings of ICALP '92*, volume 623 of *LNCS*, pages 685–695. Springer, 1992.

[Par81] David Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Theoretical Computer Science, 5th GI-Conference, Karlsruhe, Germany, March 23-25, 1981, Proceedings*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.