Bisimulation and Other Undecidable Equivalences for Lossy Channel Systems

Ph. Schnoebelen

Lab. Spécification & Vérification
ENS de Cachan & CNRS UMR 8643
61, av. Pdt. Wilson, 94235 Cachan Cedex France
email: phs@lsv.ens-cachan.fr

Abstract. Lossy channel systems are systems of finite state automata that communicate via unreliable unbounded fifo channels. Today the main open question in the theory of lossy channel systems is whether bisimulation is decidable.

We show that bisimulation, simulation, and in fact all relations between bisimulation and trace inclusion are undecidable for lossy channel systems (and for lossy vector addition systems).

1 Introduction

Channel Systems, also called *Finite State Communicating Machines*, are systems of finite state automata that communicate via asynchronous unbounded fifo channels. Fig. 1 displays an example. Channel systems are a natural model for asynchronous communication protocols and constitute the semantical basis for ISO protocol specification languages such as SDL and Estelle.

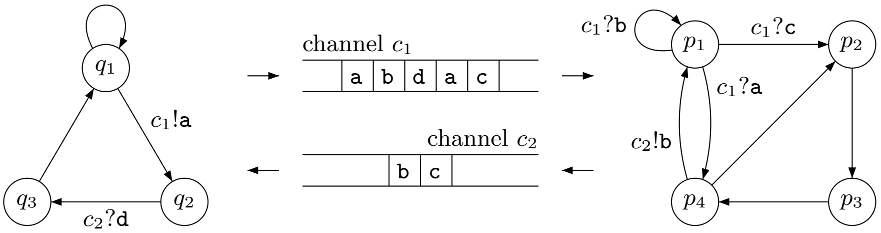

Fig. 1. A channel system with two automata and two channels

Automated verification of channel systems. Formal verification methods for channel systems are important since even the simplest communication protocols can have tricky behaviors and hard-to-find bugs. But channel systems are Turing powerful, and no verification method for them can be general and fully algorithmic. For example, existing methods only check sufficient but not necessary conditions for correctness (e.g. [JJ93]), or only terminate in some cases (e.g. [PP91]), or only deal with channel systems of a certain type (e.g. [CF97]).

Lossy channels. A few years ago, Finkel, Abdulla and Jonsson independently identified *lossy channel systems* as a very promising class of channel systems. With lossy systems, one assumes that messages can be lost while they are in transit, without any notification. Protocol designers know that unreliable channels are very real but, because they know how to cope with unreliability (e.g. with the alternating bit protocol), the classical model assumed perfect channels [Boc78,BZ81]. Therefore it is really ironic, and somewhat paradoxical, that lossy channels are "easier to analyze than perfect ones!", quoting [CFP95].

Finkel showed that termination is decidable for lossy systems [Fin94]. Abdulla and Jonsson showed the decidability of reachability, safety properties over traces, and eventuality properties over states [AJ96b]. These are fundamental results, with many practical applications in automated protocol verification.

One should not believe that lossy channel systems are trivial models where everything is decidable. First the main decidable problems do not have primitive recursive complexity [Sch01], and secondly many problems are undecidable: Abdulla and Jonsson proved that recurrent reachability properties are undecidable, so that model-checking problems for (branching-time or linear-time) temporal logic is undecidable for lossy channel systems [AJ96a], and Mayr showed that boundedness is undecidable [May00]. Also, systems where the channels are unreliable but fair (do not lose all messages all the time) cannot be analyzed [AJ96a], and one cannot say whether systems are correct with probability 1 when messages are lost with low (less than $\frac{1}{2}$) probability [ABPJ00]. Hence lossy channel systems are an example of a *partially analyzable* infinite-state system model (along with Petri nets, pushdown systems, ...) that has important practical applications [AKP97,ABJ98,AAB99].

Equivalence checking. Behavioral equivalences are a special class of verification problems where one asks whether a system S is "equivalent" to another system S' (or if two configurations of a single system are equivalent), that is whether they have the "same behavior". For such questions the first system, S, is usually a model of an implementation, while S' is a specification or a more abstract model of the same implementation. There exist several behavioral equivalences (and, more generally, implementation preorders), and one uses this or that notion depending on the situation at hand and the semantic properties of the chosen equivalence (e.g. the equivalence is a congruence). It is widely admitted that all interesting behavioral equivalences sit in van Glabbeek's *branching time – linear time spectrum* [Gla01], with bisimulation as the strongest equivalence, and trace equivalence as the weakest. For partially analyzable infinite-state systems, bisimulation is sometimes decidable [HJ99,Jan00] and sometimes not [Jan95], and other equivalences are usually at least as hard as bisimulation. Surveys can be found in [Mol96,JM99,BCMS01].

Bisimulation between lossy channel systems. Today, the main open question in the theory of lossy channel systems is the decidability of bisimulation, and other equivalences. In this direction, almost nothing is known[1]. Abdulla and Kindahl [AK95] studied equivalence problems between a lossy channel system and a finite state system, but such problems are less general than checking equivalence between two infinite systems [JKM01] and the decidability results of [AK95] cannot be used for the general case.

Our contribution. In this paper we show that all equivalences in the branching time – linear time spectrum are undecidable between two lossy channel systems. Our construction is inspired by Jančar's breakthrough result[2] and extends it. The same undecidability proof applies to bisimulation, simulation, trace inclusion, all the standard equivalences (like ready-simulation and failure equivalence), all the exotic equivalences (like 2-nested simulation and possible futures equivalences), and any equivalence or preorder (in fact, any relation) not yet invented as long as it is more discriminating than trace inclusion and less than bisimulation. The proof even shows undecidability for lossy vector addition systems, a weaker model that one uses when channels are not fifo [BM99].

Plan of the paper. Section 2 recalls basic notions (words, transition systems, behavioral equivalences). We define channel systems (extended, standard, and lossy) in section 3. The main results, given in section 4, are stated in terms of so called "front-lossy" systems, and section 5 considers the situation with "classic-lossy" systems.

2 Basic notions

2.1 Words and the subword relation

Given a finite alphabet $\Sigma = \{a, b, \ldots\}$, we let $\Sigma^* = \{u, v, \ldots\}$ denote the set of all finite words over Σ. For $u, v \in \Sigma^*$, we write $u.v$ (also uv) for the *concatenation* of u and v. We write ε for the empty word and Σ^+ for $\Sigma^* \setminus \{\varepsilon\}$. The length of $u \in \Sigma^*$ is denoted $|u|$.

The *subword relation*, denoted $u \sqsubseteq v$, relates any two words u and v s.t. u can be obtained by erasing some (possibly zero) letters from v, i.e. when u is some $a_1 \ldots a_n$, v is some $b_1 \ldots b_m$ and there are indexes $1 \leq k_1 < k_2 < \cdots < k_n \leq m$ s.t. $a_i = b_{k_i}$ for all $i = 1, \ldots, n$. For example $\mathtt{abba} \sqsubseteq \underline{\mathtt{ab}}\mathtt{racada}\underline{\mathtt{b}}\mathtt{r}\underline{\mathtt{a}}$, as we explain by underlining the b_{k_i}s. The subword relation is a partial ordering, with ε as least element, and compatible with concatenation: $u \sqsubseteq v$ and $u' \sqsubseteq v'$ entail $uu' \sqsubseteq vv'$. We write $u \sqsubset v$ when $u \sqsubseteq v$ and $v \not\sqsubseteq u$, that is when $u \sqsubseteq v$ and $|u| < |v|$.

When C is a finite index set, $\Sigma^{*C} = \{U, V, \ldots\}$ is the set of mappings from C to Σ^*, i.e. the set of C-indexed tuples of words. Subword ordering, concatenation

[1] Note that undecidability of trace equivalence can be derived easily from the undecidability of boundedness, but we are not aware of any paper making this observation.

[2] The proof that all behavioral equivalences are undecidable for P/T nets [Jan95]. Proving undecidability of (bi)simulation can be difficult, and all the non-trivial cases we know of are inspired by Jančar's method.

and size extend to tuples from Σ^{*C} in the obvious way:

$$U \sqsubseteq V \overset{\text{def}}{\Leftrightarrow} \forall c \in C : U(c) \sqsubseteq V(c), \qquad (U.V)(c) \overset{\text{def}}{=} U(c).V(c) \text{ for any } c \in C,$$
$$U \sqsubset V \overset{\text{def}}{\Leftrightarrow} U \sqsubseteq V \text{ and } V \not\sqsubseteq U, \qquad |U| \overset{\text{def}}{=} \sum_{c \in C} |U(c)|.$$

We abuse notation and write ε for any tuple of empty words.

For finite sets X (like Σ and C) we write $|X|$ to denote the cardinal of X.

2.2 Labeled transition systems and behavioral equivalences

A *labeled transition system* (a LTS) is a triple $\langle Conf, \Gamma, \rightarrow \rangle$ where $Conf = \{s, t, \ldots\}$ is a set of *configurations* (or states), $\Gamma = \{\alpha, \beta, \ldots\}$ is a set of *labels* and $\rightarrow \subseteq Conf \times \Gamma \times Conf$ is a set of *labeled transitions*. LTSs are used as models for the behavior of systems and usually *Conf* (and sometimes Γ) are infinite.

We write $s \xrightarrow{\alpha} s'$ for $(s, \alpha, s') \in \rightarrow$. As is usual with LTSs, we rely on a lot of special notations for sequences of transitions:

- For $\sigma \in \Gamma^*$, $s \xrightarrow{\sigma} s'$ iff $\sigma = \varepsilon$ and $s = s'$ or $\sigma = \alpha\sigma'$ and there is a s'' s.t. $s \xrightarrow{\alpha} s'' \xrightarrow{\sigma'} s'$.
- $s \xrightarrow{*} s'$ (resp. $\xrightarrow{+}$) $\overset{\text{def}}{\Leftrightarrow}$ $s \xrightarrow{\sigma} s'$ for some $\sigma \in \Gamma^*$ (resp. $\sigma \in \Gamma^+$).

We write $s \xrightarrow{\sigma}$ when $s \xrightarrow{\sigma} s'$ for some s' and say σ is a *trace* (sometimes called "prefix trace" to distinguish from maximal traces) of s.

For our purpose we only need define bisimulation and trace equivalence (and inclusion). We refer to [Gla01] for more definitions and motivations of the behavioral equivalences.

Trace equivalence: We write $s \subseteq_{\mathrm{Tr}} t$ when all traces of s are traces of t, i.e. when $s \xrightarrow{\sigma}$ implies $t \xrightarrow{\sigma}$ for all $\sigma \in \Gamma^*$. We write $s =_{\mathrm{Tr}} t$ when $s \subseteq_{\mathrm{Tr}} t \subseteq_{\mathrm{Tr}} s$ and say s and t are *trace-equivalent*.

Bisimulation: A relation $R \subseteq Conf \times Conf$ between the configurations of some LTS is a *simulation* if it has the *transfer property*, i.e. if for any pair $(s, t) \in R$ and any step $s \xrightarrow{\alpha} s'$, there is a $t \xrightarrow{\alpha} t'$ s.t. $(s', t') \in R$. R is a *bisimulation* if it has the transfer property both ways, i.e. if R and R^{-1} are simulations. Two configurations s and t are *bisimilar*, written $s \sim t$, if $(s, t) \in R$ for some bisimulation R.

3 Channel systems

A channel system combines several finite automata but here we restrict to *one single* automaton[3]. However, we introduce *extended* channel systems, which are

[3] This is no loss of generality since one can always safely replace several automata by a single product automaton, perhaps at the cost of an exponential blowup but

channel systems extended with the possibility of testing a channel for emptiness (a test standard channel systems cannot do).

Definition 3.1 (Extended channel system). *An* extended channel system *is a tuple* $S = \langle Q, \Sigma, C, \Gamma, \Delta, \Theta\rangle$ *where*
– $Q = \{q, p, \ldots\}$ *is a finite set of* control states*,*
– $\Sigma = \{a, b, \ldots\}$ *is a finite alphabet of* messages*,*
– $C = \{c_1, \ldots, c_m\}$ *is a finite set of* channels*,*
– $\Gamma = \{\alpha, \beta, \ldots\}$ *is a finite alphabet of* labels*,*
– $\Delta \subseteq Q \times \Sigma^{*C} \times \Gamma \times Q \times \Sigma^{*C}$ *is a finite set of* standard rules*, and*
– $\Theta \subseteq Q \times C \times \Gamma \times Q$ *is a finite set of* extended rules*.*

A *configuration* of S is a pair $(q, W) \in Q \times \Sigma^{*C}$ where q is the current control state of the system, and W is the current content of the channels: $W(c) = u$ means that c contains u.

A standard rule of the form $\langle q, U, \alpha, q', V\rangle$, written $\langle q, ?U, \alpha, q', !V\rangle$ for clarity, means that S can move from q to q' by reading U from the front of its channels, and then writing V to their tails. For a rule $\langle q, ?U, \alpha, q', !V\rangle$, we may omit writing $?U$ (resp. $!V$) when $U = \varepsilon$ (resp. $V = \varepsilon$).

An extended rule of the form $\langle q, c, \alpha, q'\rangle$, written $\langle q, c = \varepsilon?, \alpha, q'\rangle$ for clarity, means that S can move from q to q' if c is empty.

The corresponding steps give rise to visible label α. This is formalized by associating a labeled transition system with S:

Definition 3.2 (Behavior of channel systems). *For a channel system* $S = \langle Q, \Sigma, C, \Gamma, \Delta, \Theta\rangle$*, the labeled transition system* $\mathcal{S}_S = \langle \mathit{Conf}_S, \Gamma, \rightarrow\rangle$ *associated with* S *is defined by*
– $\mathit{Conf}_S = Q \times \Sigma^{*C}$*, and*
– $(q, W) \xrightarrow{\alpha} (q', W')$ *iff (1) there exists a standard rule* $\langle q, ?U, \alpha, q', !V\rangle \in \Delta$ *and some* $W'' \in \Sigma^{*C}$ *s.t.* $W = U.W''$ *and* $W' = W''.V$*, or (2)* $W' = W$ *and there exists an extended rule* $\langle q, c = \varepsilon?, \alpha, q'\rangle \in \Theta$ *s.t.* $W(c) = \varepsilon$*.*

3.1 Standard channel systems and other restrictions

We say S is a *standard channel system* when Θ is empty (no emptiness tests) and then simply write $S = \langle Q, \Sigma, C, \Gamma, \Delta\rangle$.

We say S is a *counter machine* (also a Minsky machine) when Σ contains one single message (say, $\Sigma = \{\mathtt{1}\}$). Then the channels are called counters since they behave like registers containing numbers. For a counter machine, we may replace words $u, v \in \{\mathtt{1}\}^*$ by numbers and write e.g. $W(c) = 3$ and $c = 0?$ for $W(c) = \mathtt{1}^3$ and $c = \varepsilon?$.

We say S is a *vector addition systems with states* (a VASS) when it is both a counter machine and a standard channel system, i.e. a counter machine without

this paper is only concerned with (un)decidability issues. Replacing several communicating automata with one single global automaton also has the consequence that the global automaton ends up sending messages to itself, in which case it is more natural to think in term of fifo *buffers* rather than channels.

zero-test. VASSes (and counter machines) can be used to model channel systems where the channels are not fifo, i.e. where messages can be read in any order.

We say S is an *m-channel(s) system* if C has m elements. For a 1-channel system, we may omit the name of the single channel and write standard rules as $\langle q, ?u, \alpha, q', !v\rangle$ (where $u, v \in \Sigma^*$). Note that a 0-channel system S is just a classical FSA, and is its own $\mathcal{S}_S$!

We say S is *unlabeled* when $|\Gamma| = 1$. For an unlabeled S, we sometimes omit writing Γ in S and α in the rules.

3.2 Undecidability of reachability

Extended channel systems have undecidable reachability problems for two reasons: (1) standard rules can be used to simulate the tape of a Turing machine on a single fifo channel as soon as $|\Sigma| \geq 2$ [BZ81, p. 31], and (2) extended rules allow one to simulate a Turing machine on a 2-counters machine [SS63].

For our purposes we introduce the following problem, a variant of the halting problem that makes the reduction in section 4 smoother:

Non-empty Reachability.
Instance: An extended channel system S with two designated states $q, q' \in Q$,
Question: Is there some $W \in \Sigma^{*C}$ such that $W \neq \varepsilon$ and $(q, \varepsilon) \xrightarrow{*} (q', W)$?

Theorem 3.3 ([BZ81,SS63]). *Non-empty reachability is undecidable even if we restrict to one of the following two cases:*

1. *S is a standard 1-channel system,*
2. *S is a 2-counters machine.*

3.3 Lossy channel systems

We now define the behavior of lossy channel systems. Note that only *standard* channel systems are considered under a lossy point of view. One could investigate lossy extended channel systems, for which reachability can be proven decidable along the lines of [May98,BM99,May00], but this is not our purpose here.

Modeling lossy channels can be done in several ways. The early way was to model a non-ideal channel by inserting an additional automaton that corrupts messages passing through it [ZWR$^+$80]. This led Finkel to his definition of *completely specified protocols*, where lossiness is modeled by adding to Δ all rules of the form $\langle q, c?a, q\rangle$ for all $q \in Q$, $c \in C$ and $a \in \Sigma$ [Fin94].

Following Abdulla and Jonsson [AJ96b,AJ96a], we prefer to see lossy systems as systems with an altered semantics (rather than altering Δ). It turns out there are several natural possibilities:

Classic Lossiness: one assumes that any number of messages from anywhere in the channels can be lost at any time.

Front Lossiness: one assumes that messages are lost at the front of the channels (or, equivalently, while the system attempts to read them).

These are the two main proposals one finds in the literature (but each of them comes in several variants). The relative merits of these several semantics have never been discussed in the literature (even in [CFP95] where both Front Lossiness and Classic Lossiness are considered). It is not clear which proposal better fits the real world. Certainly, the Classic Lossiness semantics is mathematically more elegant, and the Front Lossiness semantics mimics Finkel's completely specified protocols.

In fact, for all practical purposes, these semantics and their variants are equivalent in the sense that a lossy channel system terminates (or is bounded, or has a given trace set, or recurrently visits a given control state) under one semantics iff it does under the other. Informally, the reason is that if a message is lost and you cannot help it, nor know it, then when and where that happens makes no difference.

Unfortunately, this sensible line of reasoning does not extend to branching-time notions of behavior, where the timing of non-deterministic choices is important. The consequence is that two configurations can be bisimilar under one lossy semantics and not under the other [GV93]. Because of this, we consider both semantics and prove undecidability of bisimulation under each one.

Definition 3.4 (Two Lossy Semantics). *With a channel system $S = \langle Q, \Sigma, C, \Gamma, \Delta\rangle$ we associate two labeled transition systems $\mathcal{L}_S = \langle Conf_S, \Gamma, \rightarrow_{\rm l}\rangle$ (Classic Lossy), $\mathcal{FL}_S = \langle Conf_S, \Gamma, \rightarrow_{\rm f}\rangle$ (Front Lossy) that only differ from $\mathcal{S}_S$ by the labeled transitions:*
– $(q, W) \xrightarrow{\alpha}_{\rm l} (q', W')$ iff there are some $V, V' \in \Sigma^{*C}$ s.t. $V \sqsubseteq W$, $W' \sqsubseteq V'$, and $(q, V) \xrightarrow{\alpha} (q', V')$ is a step in $\mathcal{S}_S$.
– $(q, W) \xrightarrow{\alpha}_{\rm f} (q', W')$ iff there exists a rule $\langle q, ?U, \alpha, q', !V\rangle \in \Delta$ and some $U', W'' \in \Sigma^{*C}$ s.t. $U \sqsubseteq U'$, $W = U'.W''$ and $W' = W''.V$.

Observe that $\rightarrow \subseteq \rightarrow_{\rm f} \subseteq \rightarrow_{\rm l}$ and that the inclusions are strict in general.

4 Reducing from perfect to lossy systems

Our undecidability proof reduces the non-empty reachability problem for an extended channel system S to the behavioral equivalence of twin configurations in some standard channel system S' obtained from S.

Let $S = \langle Q, \Sigma, C, \Delta, \Theta, q_f\rangle$ be an arbitrary unlabeled extended channel system with a designated state $q_f \in Q$. From S we build a standard system $S' = \langle Q', \Sigma, C, \Gamma', \Delta'\rangle$ where

- Q' contains two copies of the states in Q. Formally $Q' \stackrel{\text{def}}{=} Q \times \{+, -\}$. Given $x \in \{+, -\}$, we shortly write q^x for (q, x), Q^x for $Q \times \{x\}$. For $q^x \in Q'$, the value of x is called the *polarity* of q^x.
- $\Gamma' \stackrel{\text{def}}{=} \Delta \cup \Theta \cup \{\#\}$ has one label for every rule in S, plus an extra label #.

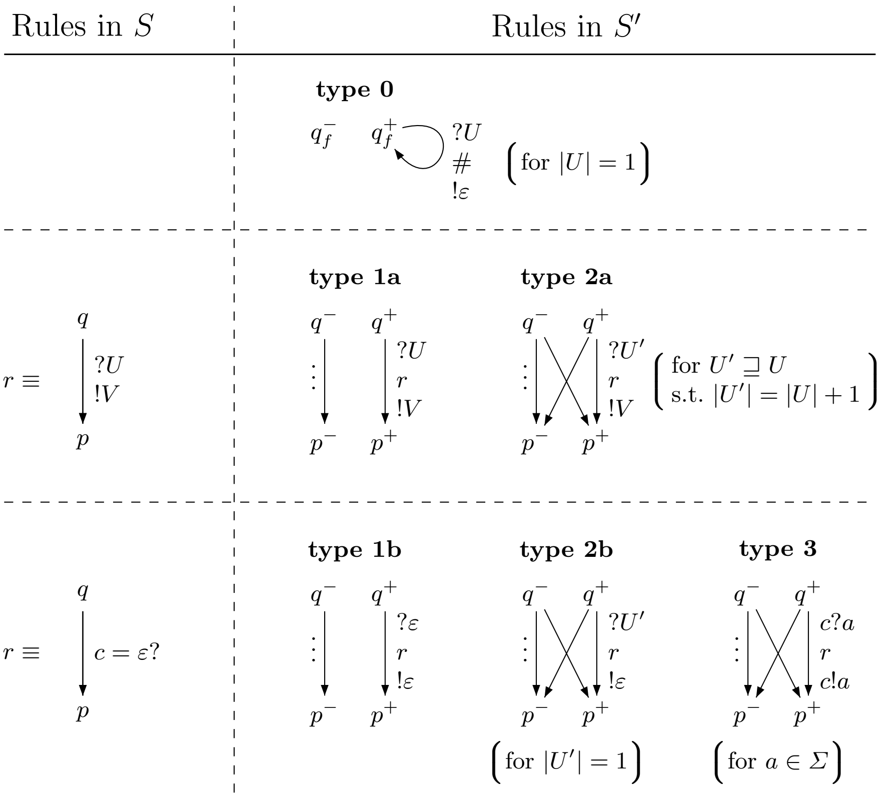

Fig. 2. Lossy system S' associated with perfect system S

There remains to define Δ', the rules of S'. These rules are all rules of one of the following type:

type 0: For any U s.t. $|U| = 1$, Δ' has a rule $\langle q_f^+, ?U, \#, q_f^+, !\varepsilon\rangle$.

type 1a: If $r = \langle q, ?U, p, !V\rangle \in \Delta$ and $x \in \{+, -\}$, then Δ' has a rule $\langle q^x, ?U, r, p^x, !V\rangle$.

type 1b: If $r = \langle q, c = \varepsilon?, p\rangle \in \Theta$ and $x \in \{+, -\}$, then Δ' has a rule $\langle q^x, ?\varepsilon, r, p^x, !\varepsilon\rangle$.

type 2a: If $r = \langle q, ?U, p, !V\rangle \in \Delta$, $x, y \in \{+, -\}$ and U' is any vector s.t. $U \sqsubseteq U'$ and $|U'| = |U| + 1$, then Δ' has a rule $\langle q^x, ?U', r, p^y, !V\rangle$.

type 2b: If $r = \langle q, c = \varepsilon?, p\rangle \in \Theta$, $x, y \in \{+, -\}$ and U' is any vector s.t. $|U'| = 1$, then Δ' has a rule $\langle q^x, ?U', r, p^y, !\varepsilon\rangle$.

type 3: If $r = \langle q, c = \varepsilon?, p\rangle \in \Theta$, $x, y \in \{+, -\}$, and $a \in \Sigma$, then Δ' has a rule $\langle q^x, c?a, r, p^y, c!a\rangle$.

A type 1 rule is type 1+ (resp. 1−) if $x = +$ (resp. $x = -$).

The construction is illustrated on Fig. 2. The intuition behind the rules of S' is the following: the behavior of S from some (q, W) can be imitated perfectly by S' from (q^+, W) and from (q^-, W) with the type 1 rules. Note that S' does not have extended rules, but this does not forbid imitating S.

Imitating from (q^+, W) only uses control states from Q^+, and imitating from (q^-, W) only uses Q^-. States from Q^+ and Q^- can be distinguished in q_f where type 0 rules exist for q_f^+ only : these rules consume from the channels, emitting one # per message, and allow to count how many messages were in the channels.

It is possible to "cross" (to move from Q^+ to Q^-, or the other way around) by a type 2 rule or a type 3 rule, but these crossing moves do not imitate moves of S. Indeed, using a type 2 rule consumes one more message from the channel (compared to the corresponding type 1 rule) and using a type 3 rule assumes c is not empty when S tests c for emptiness.

Some behavioral properties of S, seen as an error-free channel system, can be expressed as properties of S', seen as a lossy system: say that a configuration (q, W) is *bad* if it allows reaching q_f on non-empty channels, i.e. if $(q, W) \xrightarrow{*} (q_f, W')$ in $\mathcal{S}_S$ for some $W' \neq \varepsilon$. We say (q, W) is *good* when it is not bad. The predecessors of a bad configuration are bad, so the successors of a good configuration are good.

The next two lemmas say that bad configurations in $\mathcal{S}_S$ give rise in $\mathcal{FL}_{S'}$ to twin configurations with different traces, while good configurations give rise to bisimilar twin configurations.

Lemma 4.1. *If (q, W) is bad in $\mathcal{S}_S$, then $(q^+, W) \not\subseteq_{\mathrm{Tr}} (q^-, W)$ in $\mathcal{FL}_{S'}$.*

Proof. We just sketch the idea and refer to appendix A.1 for the detailed proof.

If $\mathcal{S}_S$ has a run $(q, W) \xrightarrow{*} (q_f, W')$ where $|W'| = k > 0$ and the run uses rules $r_1 \ldots r_m$ in that order, then $\mathcal{FL}_{S'}$ has a run of the form:

$$(q_0^+, W_0) \xrightarrow{r_1 \ldots r_m} (q_f^+, W') \xrightarrow{\#^k} (q_f^+, \varepsilon) \tag{1}$$

Now (q^-, W) can only display the same trace by crossing at some time (only q_f^+ can issue #) and this requires consuming one more message than in (1). Thus $(q^-, W) \xrightarrow{r_1 \ldots r_m} (q_f^+, W'')$ implies $|W''| < k$ and $r_1 \ldots r_m \#^k$ is not a trace of (q^-, W). □

Lemma 4.2. *Assume S is a standard channel system, or a counter machine. Let $R = \{(q^+, W), (q^-, W) \mid (q, W)$ good in $\mathcal{S}_S\}$. Then $R \cup Id_{Conf}$ is a bisimulation for $\mathcal{FL}_{S'}$.*

Proof. We just sketch the idea and refer to appendix A.2 for the detailed proof.

When the attacker plays in $\mathcal{FL}_{S'}$ a move $(q_1^x, W_1) \xrightarrow{r} (q_2^x, W_2)$ that exists in $\mathcal{S}_S$, the defender plays the corresponding move from (q_1^{-x}, W_1) and is safe since the game remains in good configurations where one cannot use the type 0 rule. If the attacker plays a move that does not exist in $\mathcal{S}_S$ (e.g. because some messages are lost, or because a type 2 or 3 rule is used) then the defender can cross and reach the attacker's configuration (perhaps using a different rule and losing messages). □

Corollary 4.3. *When S is a standard channel system or a counter machine, For any configuration (q, W), the following are equivalent:*

1. *(q, W) is a good configuration of $\mathcal{S}_S$,*
2. *(q^+, W) and (q^-, W) are bisimilar configurations of $\mathcal{FL}_{S'}$,*
3. *$(q^+, W) \subseteq_{\mathrm{Tr}} (q^-, W)$ in $\mathcal{FL}_{S'}$.*

Proof. *(3.)* implies *(1.)* (Lemma 4.1) and *(1.)* implies *(2.)* (Lemma 4.2). *(2.)* implies *(3.)* since bisimilarity entails trace equivalence. □

Let now ρ be any relation between configurations of labeled transition systems, that sits between bisimulation and trace inclusion. Corollary 4.3 yields:

Lemma 4.4. *Let S be a standard channel system or a counter machine with a designated q_f, let (q, W) be any configuration of S. Then (q, W) is good in $\mathcal{S}_S$ iff $(q^+, W)\rho(q^-, W)$ in $\mathcal{FL}_{S'}$.*

We can now state our main result with the following two theorems:

Theorem 4.5 (Undecidability for front-lossy systems). *Any relation ρ in the branching time – linear time spectrum is undecidable between configurations of front-lossy channel systems, even if we restrict to 1-channel systems, or to VASSes.*

Proof. Combine Theorem 3.3 and Lemma 4.4. □

Theorem 4.6 (Undecidability for classic-lossy systems). *Any relation ρ in the branching time – linear time spectrum is undecidable between configurations of classic-lossy channel systems, even if we restrict to VASSes.*

Proof (Idea). We would like to say that $\mathcal{L}_S$ and $\mathcal{FL}_S$ coincide (i.e., $\rightarrow_{\mathrm{l}} = \rightarrow_{\mathrm{f}}$) when S is a VASS, so that Theorem 4.6 is a consequence of Theorem 4.5. But this is not exactly true since front-lossy systems cannot lose what a rule just wrote into the channel. So we have to show that Lemmas 4.1 and 4.2 still hold for classic-lossy systems: the proof stays unchanged for Lemma 4.1 and only needs minor changes for Lemma 4.2. □

5 Classic-lossy systems with one channel

Our proof does not apply to classic-lossy systems with one channel: the difficulty is that a losing move in some $(q^?, W)$ cannot be punished by a crossing move in the twin configuration since rules for crossing moves can only consume from the head of the channels.

The decidability of bisimulation for these systems is the main remaining open problem. We have no strong argument that would favor a conjecture of decidability or of undecidability.

Of course, for equivalences that are weaker than bisimulation, undecidability is easy to prove, and we can state the following two results:

Theorem 5.1. *Any relation ρ between trace inclusion and* simulation *equivalence is undecidable between configurations of classic-lossy single-channel systems.*

Proof (Idea). This uses a slight modification of our construction. When a step from (q^x, W) loses message(s), the twin configuration follows by not losing (unless this is necessary for firing the same transition). In effect, this relies on the idea that postponing losses allows more behaviors, and is compatible with the simulation preorder.

Theorem 5.2. *Any relation ρ between τ-trace inclusion and τ-bisimulation (a.k.a. observational equivalence) is undecidable between configurations of classic-lossy single-channel systems.*

Proof (Idea). This uses another slight modification of our construction. Here, when (q^x, W) loses some message(s), the twin configuration can use τ-steps to rotate the buffer, putting the lost message at the head of the buffer, then consuming it by a crossing rule, and rotating the buffer back in place.

6 Conclusion

We showed undecidability of all behavioral relations over lossy channel systems, where a behavioral relation is any relation that sits between bisimulation and trace inclusion (the two endpoints of van Glabbeek's branching time – linear time spectrum [Gla01]).

The proof also applies to lossy vector addition systems. It can deal with the front-lossy semantics (as soon as we have one channel or two counters) and the classic-lossy semantics (as soon as we have two counters). Note that with only one counter, bisimulation is decidable even for perfect systems [Jan00].

The proof *does not apply* to systems with one single channel under the classic-lossy semantics, and this is the main direction we see for future work. Another case where the proof does not apply is when we restrict to systems where the rules $\langle q, ?U, q', !V\rangle$ satisfy $|UV| \leq 1$ (at most one message is consumed/produced at a time).

A Technical appendix

A.1 Proof of Lemma 4.1

Assume (q, W) is bad, i.e. $\mathcal{S}_S$ has a run of the form $(q, W) \xrightarrow{*} (q_f, W')$ where $|W'|$ is some $k > 0$. Assume this run has length m and uses rules $r_1 \dots r_m$ in that order.

S' can mimic this run. Formally, S' has a run (of length $m+k$) of the following form:

$$(q_0^+, W_0) \xrightarrow{r_1} (q_1^+, W_1) \xrightarrow{r_2} \cdots \xrightarrow{r_m} (q_m^+, W_m) \xrightarrow{\#} \dots \xrightarrow{\#} (q_{m+k}^+, \varepsilon) \qquad (2)$$

where $(q_0^+, W_0) = (q^+, W)$, $(q_m^+, W_m) = (q_f^+, W')$ and $q_{m+j}^+ = q_f^+$ for all $j = 0, \ldots, k$. Run (2) first uses type 1+ rules (m times), then type 0 rules (k times). Since no message is lost, this is at the same time a run of $\mathcal{S}_{S'}$, a run of $\mathcal{FL}_{S'}$, and a run of $\mathcal{L}_{S'}$.

Write σ for $r_1 \ldots r_m.\#^k$. Run (2) proves that σ is a trace of (q^+, W) in $\mathcal{FL}_{S'}$. We show σ is not a trace of (q^-, W): for this we assume, by way of contradiction, that $\mathcal{FL}_{S'}$ has a run

$$(p_0, V_0) \xrightarrow{r_1}_{\mathrm{f}} (p_1, V_1) \xrightarrow{r_2}_{\mathrm{f}} \cdots \xrightarrow{r_m}_{\mathrm{f}} (p_m, V_m) \xrightarrow{\#}_{\mathrm{f}} (p_{m+1}, V_{m+1}) \ldots \xrightarrow{\#}_{\mathrm{f}} (p_{m+k}, V_{m+k}) \tag{3}$$

with $(p_0, V_0) = (q^-, W)$. We derive a contradiction in a few easy lemmas.

Lemma A.1. *If $0 \leq i \leq m$ then $p_i = q_i^+$ or $p_i = q_i^-$. If $m \leq i \leq m + k$ then $p_i = q_f^+$.*

Proof. Obvious since the label (r_i or $\#$) of the step taken in (p_i, V_i) uniquely determines what are the possible values for p_i (here we rely on $k > 0$). □

One can relate the i-th step $(q_{i-1}^+, W_{i-1}) \rightarrow (q_i^+, W_i)$ of run (2) and the i-th step $(p_{i-1}, V_{i-1}) \rightarrow_{\mathrm{f}} (p_i, V_i)$ of run (3) by the following:

Lemma A.2. *For all $1 \leq i \leq m$ and all $c \in C$*

$$|W_i(c)| \geq |V_i(c)|, \tag{4}$$

$$|W_i(c)| - |W_{i-1}(c)| \geq |V_i(c)| - |V_{i-1}(c)|. \tag{5}$$

Furthermore, if p_i and p_{i-1} have different polarities, then

$$|W_i| - |W_{i-1}| > |V_i| - |V_{i-1}|. \tag{6}$$

Proof. The proof is by induction over i.

We first prove (5) and (6) assuming (4) holds for $i - 1$ (note that (4) holds for $i = 0$ since $W_0 = W = V_0$).

The i-th step of run (2) uses the type 1+ rule associated with $r_i \in \Delta \cup \Theta$. We have two subcases:

- If r_i is a standard rule $\langle q_{i-1}, ?U, q_i, !V \rangle$, then $|W_i(c)| = |W_{i-1}(c)| - |U(c)| + |V(c)|$ for any c (since this run is not lossy).
  Now the i-th step of run (3) may use the corresponding type 1a– rule, so that in general $|V_i(c)| \leq |V_{i-1}(c)| - |U(c)| + |V(c)|$ (since this step can be lossy) and we get (5). This step may also use a type 2a rule where one more message is consumed (notwithstanding possible losses), in which case we have both (5) and (6).
- If r_i is an extended rule $\langle q_{i-1}, c = \varepsilon?, q_i \rangle$, then $W_i = W_{i-1}$ and $W_{i-1}(c) = \varepsilon$. We deduce $V_{i-1}(c) = \varepsilon$ by ind. hyp., using (4), so that the i-th step of run (3) cannot use a type 3 rule.
  Thus that step uses a type 1b rule (and then we get $|V_i(c)| \leq |V_{i-1}(c)|$ because losses are possible, so that (5) holds), or a type 2b rule, where we further have (6) since one message must be consumed.

Note that a polarity change *requires* a type 2 rule, so that (6) holds in these cases. Once (5) is proven for i, we get (4) for i by adding (4) for $i-1$ to (5). □

Lemma A.3. *For $i = 0, \dots, m$, let n_i be the number of polarity changes in the first i steps of run (3). Then $n_i \leq |W_i| - |V_i|$.*

Proof. Easy induction over i. The base case uses $V_0 = W = W_0$ and $n_0 = 0$. The inductive step is given by Lemma A.2. □

Now we have our contradiction : since $p_0 = q^-$ and $p_m = q_f^+$ do not have the same polarity, we have $n_m > 0$, and hence $|V_m| < |W_m| = k$ by Lemma A.3. But if $|V_m| < k$, we cannot have $(p_m, V_m) \xrightarrow{\#^k}_{\mathrm{f}}$ since type 0 rules consume messages.

Thus our assumption is contradictory. There is no run of the form (3) and σ is not a trace of (q^-, W) in $\mathcal{FL}_{S'}$. Q.E.D.

A.2 Proof of Lemma 4.2

We show $R \cup Id_{Conf}$ has the transfer property in both directions. It is enough to consider a pair $(q^+, W)R(q^-, W)$ where (q, W) is a good configuration.

We first deal with the left-to-right transfer: Assume $(q^+, W) \xrightarrow{\alpha}_{\mathrm{f}} (p^x, W')$. We proceed by case analysis and consider what rule was used by that step:

a type 0 rule: This implies $q = q_f$, $W \neq \varepsilon$, and contradicts the assumption that (q, W) is good.

a type 2 or a type 3 rule: These rules allow crossing, $(q^-, W) \xrightarrow{\alpha}_{\mathrm{f}} (p^x, W')$ is possible and we are connected in Id_{Conf}.

a type 1a+ rule and the step loses no message: Then $x = +$ and $(q, W) \xrightarrow{\alpha} (p, W')$ is a step in $\mathcal{S}_S$, so that (p, W') is good. We have $(q^-, W) \xrightarrow{\alpha}_{\mathrm{f}} (p^-, W')$ using a type 1a– rule, $(p^+, W')R(p^-, W')$ and we are connected in R.

a type 1a+ rule and the step loses some message(s): Then $(q^-, W) \xrightarrow{\alpha}_{\mathrm{f}} (p^x, W')$ can be obtained with a type 2a rule (one picks the rule where U' accounts for both what the type 1a+ rule consumes and one of the lost messages) and we are connected in Id_{Conf}.

a type 1b+ rule and $W(c) = \varepsilon$ and the step loses no message: Then $x = +$ and, since c is empty, $(q, W) \xrightarrow{\alpha} (p, W')$ is a step in $\mathcal{S}_S$, so that (p, W') is good. We have $(q^-, W) \xrightarrow{\alpha}_{\mathrm{f}} (p^-, W')$ using a type 1b– rule, $(p^+, W')R(p^-, W')$ and we are connected in R.

a type 1b+ rule and the step loses some message(s): Then $(q^-, W) \xrightarrow{\alpha}_{\mathrm{f}} (p^x, W')$ can be obtained with a type 2b rule.

a type 1b+ rule and $W(c) \neq \varepsilon$ and the step loses no message: Then $(q^-, W) \xrightarrow{\alpha}_{\mathrm{f}} (p^x, W')$ can be obtained with a type 3 rule. This rule consumes some message a at the front of c and inserts it at the back, but Lemma 4.2 only applies to counter machines or standard systems, and the use of a type 1b rule implies S' is a VASS.

The right-to-left part of the transfer property is similar.

References

[AAB99] P. A. Abdulla, A. Annichini, and A. Bouajjani. Symbolic verification of lossy channel systems: Application to the bounded retransmission protocol. In *Proc. 5th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99), Amsterdam, The Netherlands, March 1999*, volume 1579 of *Lecture Notes in Computer Science*, pages 208–222. Springer, 1999.

[ABJ98] P. A. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy FIFO channels. In *Proc. 10th Int. Conf. Computer Aided Verification (CAV'98), Vancouver, BC, Canada, June-July 1998*, volume 1427 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 1998.

[ABPJ00] P. A. Abdulla, C. Baier, S. Purushothaman Iyer, and B. Jonsson. Reasoning about probabilistic lossy channel systems. In *Proc. 11th Int. Conf. Concurrency Theory (CONCUR'2000), University Park, PA, USA, Aug. 2000*, volume 1877 of *Lecture Notes in Computer Science*, pages 320–333. Springer, 2000.

[AJ96a] P. A. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *Information and Computation*, 130(1):71–90, 1996.

[AJ96b] P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

[AK95] P. A. Abdulla and M. Kindahl. Decidability of simulation and bisimulation between lossy channel systems and finite state systems. In *Proc. 6th Int. Conf. Theory of Concurrency (CONCUR'95), Philadelphia, PA, USA, Aug. 1995*, volume 962 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 1995. Long version available at `http://www.docs.uu.se/docs/avds/`.

[AKP97] P. A. Abdulla, M. Kindahl, and D. Peled. An improved search strategy for lossy channel systems. In *Proc. Joint Int. Conf. Formal Description Techniques and Protocol Specification, Testing, and Verification (FORTE/PSTV'97), Osaka, Japan, Nov. 1997*, pages 251–264. Chapman & Hall, 1997.

[BCMS01] O. Bukart, D. Caucal, F. Moller, and B. Steffen. Verification on infinite structures. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of Process Algebra*, chapter 9, pages 545–623. Elsevier Science, 2001.

[BM99] A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. In *Proc. 16th Ann. Symp. Theoretical Aspects of Computer Science (STACS'99), Trier, Germany, Mar. 1999*, volume 1563 of *Lecture Notes in Computer Science*, pages 323–333. Springer, 1999.

[Boc78] G. von Bochmann. Finite state description of communication protocols. *Computer Networks and ISDN Systems*, 2:361–372, 1978.

[BZ81] D. Brand and P. Zafiropulo. On communicating finite-state machines. Research Report RZ 1053, IBM Zurich Research Lab., June 1981. A short version appears in J.ACM 30(2):323–342, 1983.

[CF97] G. Cécé and A. Finkel. Programs with quasi-stable channels are effectively recognizable. In *Proc. 9th Int. Conf. Computer Aided Verification (CAV'97), Haifa, Israel, June 1997*, volume 1254 of *Lecture Notes in Computer Science*, pages 304–315. Springer, 1997.

[CFP95] G. Cécé, A. Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, 1995.

[Fin94] A. Finkel. Decidability of the termination problem for completely specificied protocols. *Distributed Computing*, 7(3):129–135, 1994.

[Gla01] R. J. van Glabbeek. The linear time – branching time spectrum I. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier Science, 2001.

[GV93] R. J. van Glabbeek and F. W. Vaandrager. Modular specification of process algebras. *Theoretical Computer Science*, 113(2):293–348, 1993.

[HJ99] Y. Hirshfeld and M. Jerrum. Bisimulation equivalence is decidable for normed process algebra. In *Proc. 26th Int. Coll. Automata, Languages, and Programming (ICALP'99), Prague, Czech Republic, July 1999*, volume 1644 of *Lecture Notes in Computer Science*, pages 412–421. Springer, 1999.

[Jan95] P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148(2):281–301, 1995.

[Jan00] P. Jančar. Decidability of bisimilarity for one-counter processes. *Information and Computation*, 158(1):1–17, 2000.

[JJ93] T. Jéron and C. Jard. Testing for unboundedness of fifo channels. *Theoretical Computer Science*, 113(1):93–117, 1993.

[JKM01] P. Jančar, A. Kučera, and R. Mayr. Deciding bisimulation-like equivalences with finite-state processes. *Theoretical Computer Science*, 258(1–2):409–433, 2001.

[JM99] P. Jančar and F. Moller. Techniques for decidability and undecidability of bisimilarity. In *Proc. 10th Int. Conf. Concurrency Theory (CONCUR'99), Eindhoven, The Netherlands, Aug. 1999*, volume 1664 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 1999.

[May98] R. Mayr. Lossy counter machines. Tech. Report TUM-I9830, Institut für Informatik, TUM, Munich, Germany, October 1998.

[May00] R. Mayr. Undecidable problems in unreliable computations. In *Proc. 4th Latin American Symposium on Theoretical Informatics (LATIN'2000), Punta del Este, Uruguay, Apr. 2000*, volume 1776 of *Lecture Notes in Computer Science*, pages 377–386. Springer, 2000.

[Mol96] F. Moller. Infinite results. In *Proc. 7th Int. Conf. Concurrency Theory (CONCUR'96), Pisa, Italy, Aug. 1996*, volume 1119 of *Lecture Notes in Computer Science*, pages 195–216. Springer, 1996.

[PP91] Wuxu Peng and S. Purushothaman Iyer. Data flow analysis of communicating finite state machines. *ACM Transactions on Programming Languages and Systems*, 13(3):399–442, 1991.

[Sch01] Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. In preparation, 2001.

[SS63] J. C. Shepherdson and H. E. Sturgis. Computability of recursive functions. *Journal of the ACM*, 10(2):217–255, 1963.

[ZWR$^+$80] P. Zafiropulo, C. West, H. Rudin, D. Cowan, and D. Brand. Towards analysing and synthesizing protocols. *IEEE Transactions on Communication*, 28(4):651–661, 1980.