On decidability of simulation and bisimulation for Lossy Channel Systems

Parosh Abdullah   Mats Kindahl

Abstract

By using an infinite-state system consisting of finite-state processes communicating via unbounded FIFO channels it is possible to model link protocols like the Alternating Bit protocol , or the HDLC protocol. It is well known that most interesting verification problems are undecidable for this class of systems. We have previously shown that by altering the behaviour of the channels so that they become *lossy*, several verification problems become decidable. In this paper we develop algorithms for deciding strong bisimulation equivalence and simulation preorder. Furthermore, we show that weak bisimulation equivalence is undecidable.

1 Introduction

During the last decade, the research on methods for algorithmic verification of concurrent and parallel systems has expanded dramatically. Several methods for verifying systems with a *finite* number of states (e.g. [CES86, FM90]) has been developed and investigated. For systems with infinite state-space, algorithmic verification is more difficult. There are many classes of systems that has an infinite state space for which you want to check different properties. Examples of such systems are real-time systems that has clocks, systems with expanding structure, and system that use unbounded FIFO-channels. There are several notations for modelling systems with infinite state space, e.g. Petri nets [Pet77], CCS [Mil89], etc. Unfortunately it is often necessary to properties for these systems manually, as most interesting properties are undecidable.

In [AJ93] a class of finite-state systems that communicate via unbounded but *unreliable* FIFO channels (called *lossy channel systems*), and it was proven that the reachability problem and the eventuality problem were decidable. Lossy channel systems were further investigated in [AJ94] where it was proven that the *model checking* problem, for both *linear* and *branching* time logics, was undecidable; it was also proven that *eventuality properties with fair channels* are undecidable. In this paper we further investigate decidability issues regarding lossy channel systems. Focusing on process algebraic relations we will show that *weak bisimulation equivalence* between lossy channel systems are undecidable. We will further show that *strong bisimulation equivalence* is decidable.

Related work: There are several papers on the analysis of systems that communicate over perfect unbounded FIFO channels. Unfortunately, most verification problems for these kind of systems are undecidable since the channels can be used to simulate the tape of a Turing Machine [BZ83].

Outline: The paper is organized as follows. In section 2 we will present the basic definitions of lossy channel systems. In section 3 we will introduce bisimulation equivalence between lossy channel systems and show some properties of bisimulation equivalence for lossy channel systems, and give an algorithm for deciding bisimulation equivalence. In section 4 we will show that weak bisimulation is undecidable, if you hide the channel operations, by reducing the *Recurrent Path Problem* to the problem of deciding weak bisimulation.

2 Systems with unreliable channels

In this section we present the definition of *lossy channel systems*, systems with unreliable channels. We also define the operational semantics of lossy channel systems.

A lossy channel system consists of two parts, a finite-state *control* and a finite set of *unbounded* FIFO channels. The channels contain sequences of messages taken from a finite alphabet and the control is a finite-state labelled transition system where each label denotes one of three different kinds of operations.

a The operation that perform the visible action a,

$c!m$ the operation that send a message m on a channel c, and

$c?m$ the operation that receive a message m from a channel c.

From [AJ93] we have the following definition of lossy channel systems:

Definition 1 *A* Lossy Channel System *is a tuple* $\mathcal{L} = \langle S, s_0, A, C, M, \delta \rangle$ *where*

S *is a finite set of* control states*,*

$s_0 \in S$ *is an* initial control state*,*

A *is a finite set of* actions*,*

C *is a finite set of* channels*,*

δ *is a finite set of* transitions*, each of which is a triple of the form* $\langle s, \alpha, s' \rangle$*, where* s *and* s' *are control states, and* α *is an* operation *of the form:*

- $c!m$*, where* $c \in C$ *and* $m \in M$*,*
- $c?m$*, where* $c \in C$ *and* $m \in M$*, or*
- a*, where* $a \in A$*.*

For a set M, let M^* denote the set of all *finite* strings of elements taken from M. For $x, y \in M^*$, let $x \bullet y$ denote the concatenation of string x and y. The empty string is denoted by ε.

A *string vector* w from C to M is a mapping $C \mapsto M^*$. In addition, we let $w\,[c := x]$ denote a string vector w' such that $w'(c) = x$ and $(\forall c' \neq c)\, w'(c') = w(c')$.

To define the *operational behaviour* we formalize the intuitive behaviour of the system as a transition system with an infinite number of states. Let $\mathcal{L}$ be a lossy channel system $\langle S, s_0, A, C, M, \delta \rangle$. A *global state* of a lossy channel system is a pair $\langle s, w \rangle$ where $s \in S$ and w is a string vector from C to M.

For a lossy channel system $\mathcal{L}$ we define the relation $\longrightarrow$ as a set of triples $\langle \gamma, \alpha, \gamma' \rangle$ where γ and γ' are global states. Let $\gamma \xrightarrow{\alpha} \gamma'$ denote $\langle \gamma, \alpha, \gamma' \rangle \in \longrightarrow$. Define the set of *events Event* to be the union of A, and the set of all $c!m$ and $c?m$, where $c \in C$ and $m \in M$.

Definition 2 *For a lossy channel system* $\mathcal{L} = \langle S, s_0, A, C, M, \delta \rangle$*, define* $\longrightarrow$ *to be the smallest trinary relation over global states such that:*

1. *if* $\langle s, c!m, s' \rangle \in \delta$*, then* $\langle s, w \rangle \xrightarrow{c!m} \langle s', w\,[c := w(c) \bullet m] \rangle$*, i.e. the control state changes from* s *to* s' *and* m *is appended to the end of channel* c*,*

2. *if* $\langle s, c?m, s' \rangle \in \delta$*, then* $\langle s, w\,[c := m \bullet w(c)] \rangle \xrightarrow{c?m} \langle s', w \rangle$*, i.e. the control state changes from* s *to* s' *and* m *is removed from the head of channel* c*,*

3. *if* $w(c) = m \bullet y$*, then* $\langle s, w \rangle \xrightarrow{\tau} \langle s, w\,[c := y] \rangle$*, i.e. the first message* m *in channel* c *is lost without changing the control state, and*

4. *if* $\langle s, a, s' \rangle \in \delta$*, then* $\langle s, w \rangle \xrightarrow{a} \langle s', w \rangle$*, i.e. the control state changes from* s *to* s' *while the action* $a \in A$ *is performed. The contents of the channels are unchanged.*

Note that this definition of $\longrightarrow$ differs from the definition in [AJ93] in that messages are only lost at the beginning of a channel, and that channel operations are visible to the environment.

For $x_1, x_2 \in M^*$, let $x_1 \preceq x_2$ denote that x_1 is a (not necessarily contiguous) substring of x_2. If w_1, w_2 are string vectors from C to M, then $w_1 \preceq w_2$ denotes that $w_1(c) \preceq w_2(c)$ for all $c \in C$. Let $\langle s_1, w_1\rangle \preceq \langle s_2, w_2\rangle$ denote that $s_1 = s_2$ and $w_1 \preceq w_2$.

3 Decidability results

In this section we present bisimulation equivalence and simulation preorder between lossy channel systems and prove that they are decidable by outlining an algorithm for checking bisimulation equivalence and simulation preorder. We also give argument for the correctness of the algorithms.

To simplify the proofs we limit ourself to using states with only one channel. It is straightforward to extend the proofs to systems with multiple channels. We introduce an alternative equivalence relation $\dot{\sim}$ and prove that it is equal to the standard definition of bisimulation equivalence when we restrict ourself to states where the channel contents are equal.

The definition of bisimulation equivalence between lossy channel systems follows the definition of bisimulation equivalence found in [Mil89].

Definition 3 *The relation R is a* bisimulation *iff; for any global states γ_1 and γ_2 such that $\gamma_1 R \gamma_2$ then for any event $\alpha \in Event \cup \{\tau\}$:*

- *if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$, for some state γ_1', then there exists global state γ_2' such that $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ and $\gamma_1' R \gamma_2'$, and*
- *if $\gamma_2 \xrightarrow{\alpha} \gamma_2'$, for some state γ_2', then there exists global state γ_1' such that $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ and $\gamma_1' R \gamma_2'$.*

Two global state γ_1 and γ_2 are *bisimilar*, written $\gamma_1 \sim \gamma_2$, if there exists a relation R such that $\gamma_1 R \gamma_2$. Two control states $s_1, s_2 \in S$ are bisimilar, written $s_1 \sim s_2$, iff $\langle s_1, \varepsilon\rangle \sim \langle s_2, \varepsilon\rangle$. Further, for global states $\gamma_1 = \langle s_1, w_1\rangle$ and $\gamma_2 = \langle s_1, w_1\rangle$, $\gamma_1 \approx_c \gamma_2$ iff $w_1 = w_2$, i.e. the states have the same channel contents, but not necessarily the same control states.

An alternative characterization of bisimilarity, which we will use later, is given using the k-bisimilarity relation, which can be found in [Mil89]. Informally, two states are k-bisimilar if they behave bisimilar in the first k steps. It is also shown that for any image-finite labelled transition system graph we have $\sim = \bigcap_{k \in \mathbf{N}} \sim_k$. It is obvious that the transition graph for any LCS is image-finite.

Definition 4 *The relation R_c is a* bisimulation with equal channels *iff; for any global states γ_1 and γ_2 such that $\gamma_1 R_c \gamma_2$ and $\gamma_1 \approx_c \gamma_2$, then for any event $\alpha \in Event \cup \{\tau\}$:*

- *if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$, for some state γ_1', then there exists global state γ_2' such that $\gamma_2 \xrightarrow{\alpha} \gamma_2'$, $\gamma_1' R_c \gamma_2'$, and $\gamma_1' \approx_c \gamma_2'$, and*
- *if $\gamma_2 \xrightarrow{\alpha} \gamma_2'$, for some state γ_2', then there exists global state γ_1' such that $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ and $\gamma_1' R_c \gamma_2'$, and $\gamma_1' \approx_c \gamma_2'$.*

Two global states γ_1 and γ_2 are *bisimilar with equal channels*, written $\gamma_1 \dot{\sim} \gamma_2$, if there exists a relation R_c such that $\gamma_1 R_c \gamma_2$. Two control states $s_1, s_2 \in S$ are bisimilar, written $s_1 \dot{\sim} s_2$, iff $\langle s_1, \varepsilon\rangle \dot{\sim} \langle s_2, \varepsilon\rangle$.

To prove that we can decide bisimulation by deciding bisimulation with equal channels we prove that they are equivalent when you start with equal channels. To show this we need the following proposition:

Proposition 3.1 *For any global states γ_1 and γ_2 such that $\gamma_1 \approx_c \gamma_2$ and operation $\alpha \in Event \cup \{\tau\}$, if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ and $\gamma_2 \xrightarrow{\alpha} \gamma_2'$, then $\gamma_1' \approx_c \gamma_2'$.*

Proof. By case analysis on the transitions of $\longrightarrow$. $\square$

Lemma 3.2 *For any two global states γ_1 and γ_2 such that $\gamma_1 \approx_c \gamma_2$, then $\gamma_1 \sim \gamma_2$ iff $\gamma_1 \dot{\sim} \gamma_2$.*

Proof by induction on k. Where $\gamma_1 \sim_k \gamma_2$ iff $\gamma_1 \dot{\sim}_k \gamma_2$, for any $k \in \mathbf{N}$. For the base case ($k = 0$) we see that $\gamma_1 \sim_0 \gamma_2$ iff $\gamma_1 \dot{\sim}_0 \gamma_2$ holds trivially. For the induction case, we assume $\gamma_1 \approx_c \gamma_2 \supset (\gamma_1 \sim_k \gamma_2) = (\gamma_1 \dot{\sim}_k \gamma_2)$, $\gamma_1 \approx_c \gamma_2$, and prove $(\gamma_1 \sim_k \gamma_2) = (\gamma_1 \dot{\sim}_k \gamma_2)$ by proving each direction separately.

$\Longrightarrow$ Assume that $\gamma_1 \sim_{k+1} \gamma_2$ and $\gamma_1 \approx_c \gamma_2$. From the definition of $\sim_{k+1}$ we then get; for any event $\alpha \in Event$, whenever $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ there exists a γ_2' such that $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ and $\gamma_1' \sim_k \gamma_2'$. By proposition 3.1 we conclude that $\gamma_1' \approx_c \gamma_2'$, from which we, together with the induction hypothesis, can conclude that $\gamma_1' \dot{\sim}_k \gamma_2'$. In a similar manner we can prove the symmetric case. From the definition of $\dot{\sim}_{k+1}$ we can see that this implies $\gamma_1 \dot{\sim}_{k+1} \gamma_2$.

$\Longleftarrow$ Assume $\gamma_1 \dot{\sim}_{k+1} \gamma_2$. From the definition of $\dot{\sim}_{k+1}$ we then get; for any event $\alpha \in Event$, whenever $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ there exists a γ_2' such that $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ and $\gamma_1' \dot{\sim}_k \gamma_2'$. Observing that $\gamma_1' \dot{\sim} \gamma_2'$ implies $\gamma_1' \approx_c \gamma_2'$, and using the induction hypothesis, we can conclude that $\gamma_1' \sim_k \gamma_2'$. In a similar manner we can show the symmetric case. From the definition of $\sim_{k+1}$, we can see that this implies $\gamma_1 \sim_{k+1} \gamma_2$.

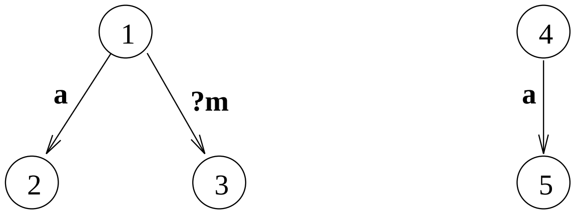

Figure 1: Counterexample for upper closure of bisimilarity

□

Bisimilarity is not upper closed, if two states are bisimilar it does not automatically mean that states that are greater are bisimilar. A trivial counterexample can be found by using the automata in figure 1 and noting that $\langle 1, \varepsilon \rangle \dot{\sim} \langle 4, \varepsilon \rangle$. Instead we can prove that *non-bisimilarity* is upper closed. If two states are non-bisimilar, then all greater states are also non-bisimilar. This can be formalized in the following lemma:

Lemma 3.3 *For global states $\gamma_1 = \langle s_1, w \rangle$ and $\gamma_2 = \langle s_2, w \rangle$, if $\gamma_1 \not\dot{\sim} \gamma_2$, then for any state γ_1' and γ_2' such that $\gamma_1 \preceq \gamma_1'$ and $\gamma_2 \preceq \gamma_2'$, follows $\gamma_1' \not\dot{\sim} \gamma_2'$.*

Proof by contradiction. Assume that $\gamma_1 \not\dot{\sim} \gamma_2$, that there exists γ_1' and γ_2' such that $\gamma_1 \preceq \gamma_1'$, $\gamma_2 \preceq \gamma_2'$, and $\gamma_1' \dot{\sim} \gamma_2'$. From the definition of $\dot{\sim}$ we observe that there exists a global state γ_2'' such that $\gamma_2'' \preceq \gamma_2'$ and $\gamma_2'' \dot{\sim} \gamma_1$. From the definition of $\dot{\sim}$ we then know that γ_1' and γ_2' has the same channel contents w'. From the definition of $\preceq$ we know that γ_1 and γ_1' has the same control state s_1, and that γ_2 and γ_2' has the same control state s_2. From the definition of $\preceq$ we know that γ_2' and γ_2'' has the same control state s_2. From the definition of $\dot{\sim}$ we know that γ_1 and γ_2'' has the same channel contents w, from which we can conclude that $\gamma_2'' = \langle s_2, w \rangle = \gamma_2$, i.e. that $\gamma_1 \dot{\sim} \gamma_2$, which is a contradiction. □

Algorithm

The basic idea of the algorithm is to check "non-bisimilarity" and do it backwards. Starting with states that are not 1-bisimilar and working backwards to get states that are non-bisimilar.

As a convenience we let ϑ denote a pair (γ_1, γ_2) such that $\gamma_1 \approx_c \gamma_2$. Let $S_\alpha(\vartheta)$ denote the set of α-successors to bisimulation state ϑ and let $P_\alpha(\vartheta)$ denote the set of α-predecessors to bisimulation state ϑ. More formally,

$$\begin{array}{rcl} S_\alpha(\vartheta) & = & \{\, \vartheta' \mid \vartheta \xrightarrow{\alpha} \vartheta' \,\} \\ P_\alpha(\vartheta) & = & \{\, \vartheta' \mid \vartheta' \xrightarrow{\alpha} \vartheta \,\}. \end{array}$$

The algorithm accepts a lossy channel system $\mathcal{L} = \langle S, s_0, A, C, M, \delta \rangle$ and two control states c_1 and c_2 and will decide if $c_1 \stackrel{\cdot}{\sim} c_2$. W denotes the "working set", which is a set of all states that have been concluded non-bisimilar but that has not yet been processed. The set V contains all sets that have been concluded non-bisimilar but have been processed.

The steps of the algorithm to check bisimulation equivalence (simulation preorder) are:

1. Set V to the empty set.
2. Let W be the set of all $\vartheta = (\gamma_1, \gamma_2)$ such that γ_1 and γ_2 have empty channels and are not 1-bisimilar.
3. If W is empty, goto step 7, otherwise proceed with step 4 to 6.
4. Pick one state ϑ in W and remove that state from W.
5. If there are no state ϑ' in V such that $\vartheta' \preceq \vartheta$, then
   (a) remove all states ϑ' in V such that $\vartheta \preceq \vartheta'$,
   (b) add ϑ to V, and
   (c) add to W all state-pairs $\vartheta' = (\gamma_1', \gamma_2')$ in $P_\alpha(\vartheta)$ such that:
      i. there is a state γ_1'' such that $\gamma_1' \xrightarrow{\alpha} \gamma_1''$ and for all γ_2'' such that $\gamma_2' \xrightarrow{\alpha} \gamma_2''$, $(\gamma_1'', \gamma_2'') \in V$, or
      ii. there is a state γ_2'' such that $\gamma_2' \xrightarrow{\alpha} \gamma_2''$ and for all γ_1'' such that $\gamma_1' \xrightarrow{\alpha} \gamma_1''$, $(\gamma_1'', \gamma_2'') \in V$.
6. Go back to step 3.
7. If $(\langle c_1, \varepsilon \rangle, \langle c_2, \varepsilon \rangle) \in V$, return "not bisimilar", else return "bisimilar".

This algorithm will add all non-bisimilar state-pairs to the set V because: for any non-bisimilar state-pair the is always a transition sequence to a state-pair that is non-1-bisimilar, i.e. a state-pair for which it can be decided that they are not bisimilar in one transition, and these state-pairs were in the set W initially.

The reason to why the algorithm terminates is that only a finite number of state-pairs can be added to V. This can be explained as follows; whenever a new state-pair ϑ is added to V it is ensured that there is no state-pair ϑ' in V such that $\vartheta' \not\preceq \vartheta$. Each sequence of state-pairs added to V can be viewed as a sequence $\vartheta_1, \vartheta_2, \ldots$ such that $\vartheta_i \not\preceq \vartheta_j$ for all $i < j$. From *Higman's theorem* it follows that there is no such sequence that is infinite. Higman's theorem can be found in [Lot83], but a version that suits our purpose better is:

Theorem 3.4 (Higman's theorem) *Let M be a finite set. There is no infinite sequence $w_1, w_2, \ldots$ of sequences such that $w_i \not\preceq w_j$ for all $i < j$.*

It is straightforward to generalize Higman's theorem to sequences of pairs of global states.

The above algorithm can be altered to cover simulation preorder by removing step 5(c)ii and replacing step 2 with:

2b. Let W be the set of all $\vartheta = (\gamma_1, \gamma_2)$ such that γ_1 and γ_2 have empty channels and are not 1-similar.

4 Undecidability results

In this section we introduce the problem of checking weak bisimulation between two lossy channel systems and prove that it is undecidable by reducing the *Recurrent Path Problem* (RPP) to weak bisimulation. For the problems in this section we use the original $\longrightarrow$, as defined in [AJ93], where we allow any message in the channel to be lost. Also, channel operations are invisible to the environment (they appear as τ transitions).

For a lossy channel system $\mathcal{L}$, a computation π of $\mathcal{L}$ and a control state s of $\mathcal{L}$, we say that π *visits s infinitely often* if there are infinitely many occurrences of global states in π with control state c.

The definition of the Recurrent Path Problem is

Definition 5 (Recurrent Path Problem) *The definition of RPP is:*

Instance: *A lossy channel system $\mathcal{L}$ and a control state c in $\mathcal{L}$*

Question: *Is there a computation of $\mathcal{L}$ visiting c infinitely often?*

In a similar manner, we can define the problem of deciding weak bisimilarity between lossy channel systems as:

Definition 6 (Weak Bisimulation) *The definition of weak bisimulation is:*

Instance: *Two lossy channel systems $\langle S_1, s_{01}, A_1, C_1, M_1, \delta_1 \rangle$ and $\langle S_2, s_{02}, A_2, C_2, M_2, \delta_2 \rangle$.*

Question: *Is there a computation of $\mathcal{L}$ visiting c infinitely often?*

Theorem 4.1 *RPP is undecidable.*

The proof of this can be seen in [AJ94].

Theorem 4.2 *RPP is reducible to weak bisimulation.*

Proof. We are given an instance Π of RPP where we have lossy channel system $\mathcal{L} = \langle \mathcal{S}, \int_{\prime}, \mathcal{A}, \mathcal{C}, \mathcal{M}, \delta \rangle$ and a control state c. We will construct two lossy channel systems $\mathcal{L}_1$ and $\mathcal{L}_2$ such that Π has a solution if and only if $\mathcal{L}_1$ and $\mathcal{L}_2$ are weakly bisimilar.

We define $\mathcal{L}_1 = \langle S \cup \{c'\}, s_0, A \cup \{a'\}, C, M, \delta' \rangle$, where

$$\delta' = \delta - \{ \langle c_t, a, c \rangle \mid \langle c_t, a, c \rangle \in \delta \} \cup \{ \langle c_t, a, c' \rangle \mid \langle c_t, a, c \rangle \in \delta \} \cup \{ \langle c', a', c \rangle \},$$

a' is a new action not in A, and c' is a new control state not in S. We define $\mathcal{L}_2 = \langle \{t\}, t, \{a'\}, \emptyset, \emptyset, \{ \langle t, a, t \rangle \mid a \in A \cup \{a'\} \} \rangle$. $\square$

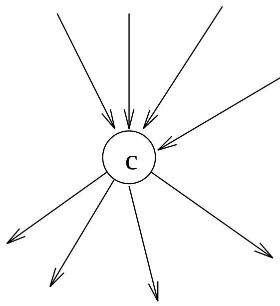

Figure 2: Instance of RPP problem

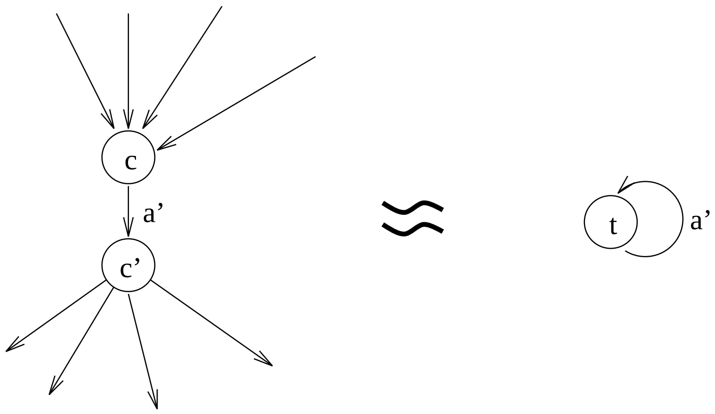

Figure 3: Weak bisimulation instance of RPP problem

On an informal level, given an instance of RPP (as in figure 2) we split the state c into two states, with a transition between them (see figure 3). These two states are visited infinitely often iff the transition between them is taken infinitely often. As the transition is labeled with a unique action a', this is equivalent to checking weak bisimulation between the modified automata and an automata that can do a' infinitely often.

5 Conclusion

In this article we have show that strong bisimulation (and simulation) between systems of finite-state processes that communicate over unbounded but lossy FIFO channels are decidable. We have also shown that weak bisimulation is undecidable, if you hide the channel operations.

Our result generalize to other types of sequences that can lose elements, e.g. "lossy stacks", "lossy tapes" (for "lossy Turing machines"). It follows that the halting problem for Turing machines with "lossy" tapes is decidable.

6 Future work

It is, as shown, possible to extend the algorithm for strong bisimulation to cover also simulation, but a more interesting alternative is to decide simulation by deciding simulation preorder between a lossy channel system and a finite-state "specification" which does *not* use FIFO channels. This would make both weak and strong simulation preorder decidable, it can be argued that many specifications usually are written in a manner which makes them finite-state, but they are used to check an infinite state system. For which simulation preorder algorithms would be useful.

Further, the model we describe here is very deterministic, messages are only lost at the beginning of a channel and every operation gives us a very predictable channel state. This high level of control unfortunately limits our modeling capacity, which is undesirable. We are currently investigating the possibility of allowing a more general model of losses, in the same manner as in [AJ93] while still maintaining bisimilarity.

Another interesting area, which we are also investigating, is if we can allow weak bisimilarity between lossy channel systems by making the channel operation visible, or by restricting one of the automata to be finite-state in the same manner as described above for simulation preorder.

References

[AJ93] Parosh Aziz Abdulla and Bengt Jonsson. Verifying programs with unreliable channels. In *Proc. 8^{th} IEEE Int. Symp. on Logic in Computer Science*, 1993.

[AJ94] Parosh Aziz Abdulla and Bengt Jonsson. Undecidable verification problems for programs with unreliable channels. Technical Report 94/47, Dept. of Computer Systems, Uppsala University, Sweden, 1994.

[BZ83] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 2(5):323–342, April 1983.

[CES86] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. on Programming Languages and Systems*, 8(2):244–263, April 1986.

[FM90] J.C. Fernandez and L. Mounier. Verifying bisimulations on the fly. In *Proc. FORTE '90*, pages 91–105, Nov. 1990.

[Lot83] M. Lothaire. *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, 1983.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[Pet77] J.L. Peterson. Petri nets. *Computing Surveys*, 9(3):221–252, 1977.