A short visit to the STS hierarchy [3]

Nathalie Bertrand [1] and Philippe Schnoebelen [2]

Lab. Spécification & Vérification, CNRS & ENS de Cachan,
61, av. Pdt. Wilson, 94235 Cachan Cedex France

Abstract

The hierarchy of Symbolic Transition Systems, introduced by Henzinger, Majumdar and Raskin, is an elegant classification tool for some families of infinite-state operational models that support some variants of a symbolic "backward closure" verification algorithm. It was first used and illustrated with families of hybrid systems.

In this paper we investigate whether the STS hierarchy can account for classical families of infinite-state systems outside of timed or hybrid systems.

Key words: Symbolic transition systems, well-structured transition systems, STS hierarchy.

1 Introduction

Verification of infinite-state systems is a very active field of research where one studies how the algorithmic techniques that underly the successful technology of model checking for finite-state systems can be extended to more expressive computational models [BCMS01]. Many different models have been studied, ranging from infinite-data models (like channel systems) to infinite-control models (like process algebras), including timed automata and hybrid systems. General undecidability results are worked around by discovering special restricted subclasses where decidability can be recovered for specific verification problems, and our understanding of the compromises between expressivity and tractability improves regularly.

There have been some attempts at bringing some order inside the existing plethora of scattered results. One way to do this is to discover conditions that (1) support some generic verification algorithms, and (2) can account

[1] Email: bertrand@lsv.ens-cachan.fr

[2] Email: phs@lsv.ens-cachan.fr

[3] This research was supported by *Persée*, a project funded by the *ACI Sécurité Informatique* of the French Ministry for Scientific Research.

for a rich enough variety of models. The *well-structured transition systems* (WSTS) of [AČJT00,FS01] are one such attempt, where the key notion is the existence of a well-quasi-order between configurations that is compatible with transitions. The WSTS idea applies widely, and instances exist in many classes of models [FS01].

The *symbolic transition systems* (STS) of [HMR05] are another attempt. Actually [HMR05] defines a *hierarchy* of five different levels: STS1 to STS5. All levels are defined in the same way: a system is STSk iff its set of configurations yields a finite quotient modulo $\approx_k$, an equivalence relation that relates states with similar "behavior". The equivalences from $\approx_1$ to $\approx_5$ are coarser and coarser, and systems in the STSk class are also in STS$(k+1)$. Additionally, five variants of a generic symbolic *closure algorithm* are given, one for each class, allowing verification of properties ranging from μ-calculus model checking (for the class STS1) to reachability properties (for the class STS5).

While the STS idea is illuminating, its weak point is that it is not widely applicable. In [HMR05], all the given examples of classes STS1 to STS5 are some restricted families of hybrid systems. And no instance of STS4 systems is given. As a consequence it is not clear whether the classification has any impact beyond hybrid and timed systems.

Our contribution. We look at well-known families of models for which verification results exist, and that are not related to hybrid systems: Petri nets, pushdown systems, and channel systems. In particular, we consider several variants of *lossy channel systems* [AJ96,CFP96]. For these families, a natural question is whether they give rise to systems sitting inside some level of the STS hierarchy.

Here we are only considering semantical issues: we ask whether a given system model with a given set of observable properties gives rise to STSk transition systems. We are not concerned with algorithmic issues and symbolic verification, even though the STS hierarchy meets its purpose when systems can be equipped with a working region algebra [HMR05].

A general outcome of our investigation is that only systems that are well-structured in the sense of [FS01] can fit in the STS hierarchy, at level STS5 (or, sometimes, STS4). Indeed, [HMR05] uses the name "*well-structured systems*" for its STS5. We argue that the close links between the two notions do not provide a perfect fit.

2 The STS hierarchy

Henzinger, Majumdar and Raskin introduced symbolic transition systems (STS) in [HMR05]. These are labeled transition systems equipped with a region algebra. However, since we do not consider algorithmic issues or symbolic verification in this paper, we will work with a simplified definition.

Definition 2.1 *A* labeled transition system *(LTS) is a tuple $\mathcal{S} = \langle S, \rightarrow, P\rangle$ where S is a (possibly infinite) set of* states, *$\rightarrow \subseteq S \times S$ is a* transition relation*, and $P \subseteq 2^S$ is a finite set of* observable properties *(or* observables*) that covers the state space: $S = \bigcup_{p\in P} p$.*

An *observation* is a set of observables. The observation $P(\sigma)$ of a state σ is $\{p \in P \mid \sigma \in p\}$.

Classically we write $\sigma \rightarrow \sigma'$ rather than $(\sigma, \sigma') \in \rightarrow$, and say that σ' is one of the successors of σ. σ is a *deadlock state* if it has no successors. A *(finite) path* in $\mathcal{S}$ is a sequence of states $\sigma_1, \cdots, \sigma_n$ such that for all i, $\sigma_i \rightarrow \sigma_{i+1}$.

The STS hierarchy is based on well-known notions of simulations and traces (see, e.g., [Gla01]).

We recall the definitions for simulations. Let $\mathcal{S} = \langle S, \rightarrow, P\rangle$ be a LTS. A binary relation $R \subseteq S \times S$ is a *simulation* on S if $\sigma R \tau$ entails:

(i) $\forall p \in P$, $\sigma \in p \Leftrightarrow \tau \in p$,

(ii) $\forall \sigma \rightarrow \sigma'$, $\exists \tau \rightarrow \tau'$ s.t. $\sigma' R \tau'$.

σ and τ are *bisimilar* –denoted by $\sigma \cong_1^{\mathcal{S}} \tau$– if there is a *symmetric* simulation R such that $\sigma R \tau$.

They are *simulation-equivalent* –denoted by $\sigma \cong_2^{\mathcal{S}} \tau$– if there are two simulations R_1 and R_2 such that $\sigma R_1 \tau$ and $\tau R_2 \sigma$.

It is well-known that bisimilarity and simulation-equivalence are equivalence relations.

We now recall the definitions for traces. Let $\mathcal{S} = \langle S, \rightarrow, P\rangle$ be a LTS. A *trace* from state σ is the observation of a path originating from σ. Formally it is a sequence $P_1 \cdots P_n$ of observations such that there exists a path $\sigma_1 \cdots \sigma_n$ with $\sigma_1 = \sigma$ and $P_i = P(\sigma_i)$ for $i = 1, \ldots, n$. Any p in P_n, the last observation along the trace, is called a *target* of the trace and we write $\sigma \xrightarrow{n} p$ when such a trace exists.

Two states σ and τ are *trace-equivalent* –denoted by $\sigma \cong_3^{\mathcal{S}} \tau$– if every trace from σ is a trace from τ, and vice-versa.

They are *distance-equivalent*, –denoted by $\sigma \cong_4^{\mathcal{S}} \tau$– if for every trace from σ with length n and target p there is a trace from τ of length n and target p, and vice versa.

They are *bounded-reach equivalent* –denoted by $\sigma \cong_5^{\mathcal{S}} \tau$– if for every trace from σ with length n and target p there is a trace from τ with length at most n and target p, and vice versa.

Clearly, trace equivalence, distance equivalence, and bounded-reach equivalence are equivalence relations.

Definition 2.2 ***(The STS hierarchy)*** **[HMR05].**
A labeled transition system $\mathcal{S} = \langle S, \rightarrow, P\rangle$ belongs to the class STSk (for $1 \leq k \leq 5$) iff the relation $\cong_k^{\mathcal{S}}$ has finite index (i.e., induces a finite number

of equivalence classes in S).

Some immediate properties of STS classes are:

Hierarchy: If $\mathcal{S}$ is in STSk, it is in STS$(k+1)$.

Finite systems: If $\mathcal{S} = \langle S, \to, P\rangle$ has finite S, then $\mathcal{S}$ is in STS1.

Trivial observables: If $\mathcal{S} = \langle S, \to, P\rangle$ has $P = \{S\}$, then $\mathcal{S}$ is in STS5. If no state is S is a deadlock state, then $\mathcal{S}$ is even in STS1.

Monotonicity w.r.t. observables: If $\mathcal{S} = \langle S, \to, P\rangle$ and $\mathcal{S}' = \langle S, \to, P'\rangle$ only differ by $P' \subseteq P$ (i.e., $\mathcal{S}$ has more observable properties than $\mathcal{S}'$), and $\mathcal{S}$ is in STSk, then $\mathcal{S}'$ too is in STSk.

3 Well-structured transition systems and the STS hierarchy

In [HMR05] the class STS5 is said to coincide with well-structured transition systems, a class of infinite-state transition systems supporting generic verification algorithms [Fin87,AČJT00,FS01]. This claim is supported by an alternative characterization of STS5 systems, using well-quasi-orderings [HMR05, Theorem 5A]. However, the link with WSTS is not made more explicit.

In this section we show that WSTS are in STS5 and consider the converse question: can any STS5 transition system be turned into a WSTS by equipping it with a "compatible" well-quasi-ordering?

We recall that a *well-quasi-ordering* (wqo) is a reflexive and transitive relation $\leq$ (over some set S) such that for any infinite sequence $x_0, x_1, \cdots$ in S, there exists indexes $i < j$ with $x_i \leq x_j$. As a consequence, a wqo is well-founded and only admits finitely many minimal elements. (In the sequel we often write, as we just did, that a set has finitely many minimal elements when we really mean "finitely many distinct minimal elements up to the equivalence induced by the wqo". This nuance is not required when the wqo is a partial ordering, i.e., is antisymmetric.)

Definition 3.1 ***(Well-Structured Transition Systems)*** **[FS01]*.***
A Well-Structured Transition System *is a transition system $\mathcal{S} = \langle S, \to, \leq\rangle$ equipped with a relation $\leq \subseteq S \times S$ which is a well-quasi-ordering (upward-) compatible with $\to$, i.e., for all $\sigma_1 \leq \tau_1$ and $\sigma_1 \to \sigma_2$ there exists $\tau_1 \to \tau_2$ with $\sigma_2 \leq \tau_2$.*

This notion of compatibility is called *strong compatibility* in [FS01]. We say $\mathcal{S} = \langle S, \to, \leq\rangle$ has *reflexive compatibility* if for all $\sigma_1 \leq \tau_1$ and $\sigma_1 \to \sigma_2$, there exists $\tau_2 \geq \sigma_2$ with either $\tau_2 = \tau_1$ or $\tau_1 \to \tau_2$ (which is denoted $\tau_1 \xrightarrow{0/1} \tau_2$ in the sequel). It is immediate that a given WSTS with strong compatibility has also reflexive compatibility.

Petri nets with k places equipped with the partial order on $\mathbb{N}^k$ are an example of well-structured transition systems (with strong compatibility), see section 4.2. Another example is the class of lossy channel systems using the subword ordering on channel contents, see section 4.3.

Definition 3.1 does not coincide with the definition used in Theorem 5A of [HMR05]. There, a well-structured system is a LTS that can be equipped with a wqo $\leq$ on the states such that for all observable properties p and $d \in \mathbb{N}$, the set of states that can reach p in less than d steps is upward-closed (a set $S' \subseteq S$ is *upward-closed* if $\sigma \in S'$ and $\sigma \leq \tau$ entail $\tau \in S'$). This is shown to coincide with STS5 systems.

Let us consider a WSTS $\mathcal{S} = \langle S, \rightarrow, \leq \rangle$ and ask whether there is a set P of observables that turn $\mathcal{S}$ into an STS5 system. Of course, setting $P = \{S\}$ works, but this does not exploit the fact that $\mathcal{S}$ is well-structured. It turns out that any set P of upward-closed observables will work, and this holds even if $\mathcal{S}$ has reflexive compatibility.

Theorem 3.2 *Let $\mathcal{S} = \langle S, \rightarrow, \leq \rangle$ be a WSTS with reflexive compatibility, and P be a finite set of upward-closed observables that covers S. Then $\langle S, \rightarrow, P \rangle$, denoted $\mathcal{S}_P$, is in STS5.*

Since [HMR05] uses a different definition, our proof of Theorem 3.2 is not a copy of the proof of [HMR05, Theorem 5A]. Moreover it is also a more direct proof since we do not deal with algorithmic aspects of predecessors computation.

Proof. For an observable $p \in P$, we let $\mathit{Orig}(p)$ denote the set of pairs $(\sigma, n) \in S \times \mathbb{N}$ such that σ can reach p within n steps: $\mathit{Orig}(p) \stackrel{\text{def}}{=} \{(\sigma, n) \mid \sigma \xrightarrow{n} p\}$. The canonical product wqo on $S \times \mathbb{N}$ is defined by

$$(\sigma, n) \sqsubseteq (\tau, m) \stackrel{\text{def}}{\Leftrightarrow} \big(\sigma \leq \tau \text{ and } n \leq m\big).$$

Let $\mathit{MinOrig}(p)$ be the set of minimal elements in $\mathit{Orig}(p)$: $\mathit{MinOrig}(p)$ is finite since $\sqsubseteq$ is a wqo. We define $\approx \subseteq S \times S$ with:

$$\sigma \approx \sigma' \stackrel{\text{def}}{\Leftrightarrow} \forall p \in P,\ \forall (\tau, m) \in \mathit{MinOrig}(p),\ \tau \leq \sigma \Leftrightarrow \tau \leq \sigma' \tag{1}$$

and claim it is a bounded-reach equivalence of finite index. That $\approx$ has finite index comes from the finiteness of $\mathit{MinOrig}(p)$. To see that it is a bounded-reach equivalence, assume $\sigma \approx \sigma'$ and $\sigma \xrightarrow{n} p$ for some $n \in \mathbb{N}$ and $p \in P$. Then $(\sigma, n) \in \mathit{Orig}(p)$ and there is some $(\tau, m) \in \mathit{MinOrig}(p)$ with $\tau \leq \sigma$ and $m \leq n$. From (1), we deduce $\tau \leq \sigma'$.

Now pick a path $\tau \rightarrow \tau_1 \rightarrow \cdots \rightarrow \tau_m$ with $\tau_m \in p$. By induction on m, and using the reflexive compatibility of $\mathcal{S}$, we show that there exist states $\sigma'_1 \cdots \sigma'_m$ such that $\sigma' \xrightarrow{0/1} \sigma'_1 \xrightarrow{0/1} \cdots \xrightarrow{0/1} \sigma'_m$ and $\tau_i \leq \sigma'_i$ for $i = 1, \ldots, m$. (Fig. 1 illustrates the proof.) Since p is upward-closed, $\tau_m \in p$ implies $\sigma'_m \in p$. Thus we have found a path witnessing $\sigma' \xrightarrow{m'} p$ for some $m' \leq m \leq n$. Hence $\approx$ is a bounded-reach equivalence and $\mathcal{S}_P$ is in STS5. $\square$

$$\begin{array}{ccc} \sigma & \approx & \sigma' \\ \big\downarrow n & & \\ p & & \end{array} \quad \Longrightarrow \quad \begin{array}{ccc} \tau & \leq & \sigma' \\ \downarrow & & \downarrow_{0/1} \\ \tau_1 & \leq & \sigma'_1 \\ \downarrow & & \downarrow_{0/1} \\ \vdots & & \vdots \\ \downarrow & & \downarrow_{0/1} \\ p \ni \tau_m & \leq & \sigma'_m \end{array}$$

Fig. 1. $\approx$ is a bounded reach equivalence

Note that, since reflexive compatibility is more general than strong compatibility, Theorem 3.2 shows a more general connection between STS5 systems and WSTS's.

A converse problem is to consider a LTS $\mathcal{S} = \langle S, \rightarrow, P \rangle$ in STS5, and try to find a well-quasi-ordering $\leq$ on S such that $\langle S, \rightarrow, \leq \rangle$ is a WSTS. Since Finkel and Schnoebelen showed that any (finitely branching) transition system could be equipped with a well-quasi-ordering $\leq$ to get a well-structured transition system [FS01], this can always be done.

However we would appreciate if the wqo that turns $\mathcal{S}$ into a WSTS were "compatible" with P. For example it would be nice if the observables in P become upward-closed sets w.r.t. the wqo since this is how P is defined in the proof of Theorem 3.2. We do not know if such a wqo can be defined for all $\mathcal{S}$ in STS5 and must leave this question open for the moment.

Remark 3.3 *Given $\mathcal{S} = \langle S, \rightarrow, P \rangle$ in STS5, [HMR05] proves that there exists a wqo on S such that, for all $p \in P$ and $d \in \mathbb{N}$, the set $\{\sigma \mid \sigma \xrightarrow{\leq d} p\}$ is upward-closed. Hence in particular every p is upward-closed (pick $d = 0$). However, the wqo they define is in general not compatible with transitions and hence does not transform $\mathcal{S}$ into a WSTS in the sense of [FS01].*

4 Looking at classical infinite-state models

All the examples of STS systems in [HMR05] are hybrid systems: timed automata, two-dimensional rectangular automata, networks of timed automata, etc. Here we study classical infinite-state systems such as pushdown automata, Petri nets and lossy channel systems and consider whether they give rise to systems in one of the STSk classes.

4.1 Pushdown automata

Pushdown automata are systems with finite control and a pushdown stack.

Formally, a pushdown automaton $PD = \langle Q, \Gamma, \Delta \rangle$ is composed of a finite set of locations Q, a stack alphabet Γ and a finite set of transition rules Δ. The rules in Δ are of the form $l \xrightarrow{\text{pop } a} l'$ or $l \xrightarrow{\text{push } a} l'$ for l, l' locations and $a \in \Gamma$. The operational semantics of PD is given as a transition system $\mathcal{S}_{PD}$ where a state (or *configuration*) has the form $\sigma = \langle l, w\rangle$ with $l \in Q$ a location and $w \in \Gamma^*$ a stack contents. We omit the obvious definition for the transitions $\sigma \to \sigma'$ (see for example [BEF+00]).

Pushdown automata are a family of infinite-state systems for which verification is relatively easy in the sense that the iterated successor relation $\xrightarrow{*}$ is recognizable and can be described by a finite transducer effectively derivable from *PD* [Cau92]. Of course there exist questions, e.g., trace equivalence, that are undecidable for these systems.

One obtains LTS's from pushdown automata by equipping the transition systems they induce with some sets of observables.

Assume $PD = \langle Q, \Gamma, \Delta\rangle$ is a pushdown automaton. The simplest and most natural observable properties are based on the locations: for each location $l \in Q$, let $p_l \stackrel{\text{def}}{=} \{\langle l, w\rangle \mid w \in \Gamma^*\}$ and $P \stackrel{\text{def}}{=} \{p_l \mid l \in Q\}$. We write PD^{l} for the class of LTS's obtained from pushdown automata with locations for observable properties.

Another option is to look at the stack and distinguish the states depending on the emptiness (or non-emptiness) of the stack. In this case there are two observable properties: $p_{\text{empty}} \stackrel{\text{def}}{=} \{\langle l, \varepsilon\rangle \mid l \in Q\}$ and $p_{\text{nempty}} \stackrel{\text{def}}{=} S \setminus p_{\text{empty}}$. This gives rise to a class of LTS's we denote PD^{s}. Finally, we write $\mathsf{PD}^{\mathsf{l,s}}$ for the class of LTS's one obtains by considering both types of observables.

Theorem 4.1 *The classes* PD^{l}, PD^{s} *and* $\mathsf{PD}^{\mathsf{l,s}}$ *give rise to LTS's that are not in STS5 in general.*

Proof. We only prove the result for PD^{l} since similar arguments work for PD^{s} (and $\mathsf{PD}^{\mathsf{l,s}}$ is dealt with using monotonicity of observables).

Consider PD_0, the pushdown automaton depicted in Fig. 2. Here from

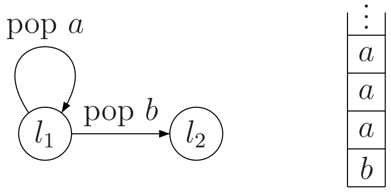

Fig. 2. PD_0, a simple pushdown automaton

location l_1, one must pop all a's before a move to location l_2 is allowed. Hence two states $\langle l_1, a^n b\rangle$ and $\langle l_1, a^m b\rangle$ are not bounded-reach equivalent unless $n = m$ (since from $\langle l_1, a^n b\rangle$ one can only reach target l_2 in $n+1$ steps). Therefore bounded-reach equivalence does not have finite index, and the STS associated with PD_0 in PD^{l} is not in STS5. □

4.2 Petri nets

We do not recall here the definition of Petri nets (see [Esp98]). Let *PN* be a Petri net with k places. Its operational semantics is given by a transition system where the states (or *markings*) are tuples from $\mathbb{N}^k$. Markings are partially ordered by the product ordering $(\mathbb{N}, \leq)^k$, or, formally

$$\langle x_1, \ldots, x_k \rangle \leq \langle y_1, \ldots, y_k \rangle \stackrel{\text{def}}{\Leftrightarrow} x_1 \leq y_1 \wedge \cdots \wedge x_k \leq y_k.$$

That $\leq$ is a wqo on $\mathbb{N}^k$ is known as Dickson's Lemma [Dic13]. For observables we consider the set of all upward closures $\uparrow m \stackrel{\text{def}}{=} \{m' \mid m \leq m'\}$ where m is a marking in $\{0, 1\}^k$. Hence an observation sees whether a place is marked or not, but does not see how many tokens are in a given place. Note that P covers S since $S = \uparrow\langle 0, \ldots, 0\rangle$. We denote by PN the class of LTS's obtained from Petri nets with the observable properties defined above.

Theorem 4.2 *The class* PN *gives rise to LTS's that are in STS*5 *but not in STS*4 *in general.*

Proof. Petri nets with $\leq$ are WSTS with strong compatibility (see [FS01] for example). A direct consequence of Theorem 3.2 is that they are STS5.

To see that they are not in STS4 in general, consider the Petri net with a single place and a single transition described in Fig. 3. Starting with n tokens,

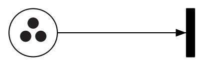

Fig. 3. A simple Petri net

the longest trace has exactly length n. Hence two different markings cannot be distance-equivalent and the distance equivalence does not have finite index on this system. □

4.3 Lossy channel systems

Several different definitions for Lossy Channel Systems (LCS) can be found in the literature: see, e.g., [Fin94,CFP96,AJ96]. In this paper we will follow the approach of Abdulla and Jonsson [AJ96] which works smoothly and is more commonly cited. For this model we introduce two variants (allowing idling or not) and consider different cases for the observables.

Definition 4.3 ***(LCS's).***
A lossy channel system $L = (Q, \mathsf{C}, \mathsf{M}, \Delta)$ *is composed of a finite set of locations* Q*, a finite set of channels* C*, a finite alphabet* M *and a finite set of transition rules* Δ*. The rules have the form* $q \xrightarrow{op} q'$ *where* q *and* q' *are locations, and* op *is an operation of the form:*

send: $c!m$ *writing message* m *to channel* c*;*

receive: $c?m$ *reading message* m *from channel* c*;*

internal action: $\surd$ *(no input/output operation).*

Operational semantics. The operational semantics of $L = (Q, \mathsf{C}, \mathsf{M}, \Delta)$ is given by a transition system where a state (or a configuration) is a pair $\langle q, w\rangle$ composed of a location q and a mapping $w : \mathsf{C} \to \mathsf{M}^*$ describing the channels contents.

The effect of an operation op on a channel contents w, denoted $op(w)$, is the channel contents w' such that:

$op = c!m$: then $w'(c) = w(c).m$ and $w'(c') = w(c')$ for $c' \neq c$;

$op = c?m$: then $m.w'(c) = w(c)$ and $w'(c') = w(c')$ for $c' \neq c$;

$op = \surd$: then $w'(c) = w(c)$ for all $c \in \mathsf{C}$.

We observe that $op(w)$ is not defined when $op = c?m$ and $w(c)$ does not start with m.

The *perfect steps* between configurations are all pairs $\langle q, w\rangle \to_{\text{perf}} \langle q', w'\rangle$ such that there is a rule $q \xrightarrow{op} q'$ in Δ with $w' = op(w)$.

Given two channels contents w and w', we write $w \sqsubseteq w'$ if w can be obtained from w' by deleting messages (whatever their place in w'). This is extended to states as follows:

$$\langle q, w\rangle \sqsubseteq \langle q', w'\rangle \overset{\text{def}}{\Leftrightarrow} q = q' \text{ and } w \sqsubseteq w'.$$

This is a wqo between states (by Higman's Lemma [Hig52]).

What we are really interested in are the *lossy steps*, obtained from perfect steps by preceding and following them by arbitrary message losses (possibly none). Formally:

$$\sigma \to_{\text{loss}} \tau \quad \overset{\text{def}}{\Leftrightarrow} \quad \exists\sigma', \exists\tau' \text{ s.t. } \sigma \sqsupseteq \sigma' \ \wedge\ \sigma' \to_{\text{perf}} \tau' \ \wedge\ \tau' \sqsupseteq \tau.$$

Idling. Starting with this definition, a natural variant is to enable idling in all configurations [BS03]. This assumption, which amounts to adding all pairs $\sigma \to \sigma$ on top of lossy steps, is a way of getting rid of deadlock states.

Observables. Natural observable properties for LCS's are associated with the locations (exactly as with pushdown automata) and we let $\mathcal{S}_L^{\text{l}}$ ("l" for "locations") denote the LTS associated in such a way with LCS L.

One may prefer to observe the contents of the channels but this requires some care in order to obtain upward-closed observables. A simple solution is to only consider upward-closed and *location-independent* properties, i.e., properties p such that for all $q, q' \in Q$ and all $w \sqsubseteq w'$, $\langle q, w\rangle \in p$ implies $\langle q', w'\rangle \in p$. For every $c \in C$, one such property is $p_c \overset{\text{def}}{=} \{\langle q, w\rangle \mid w(c) \neq \varepsilon\}$, that allows to observe (non-)emptiness of c. One obtains a set of observables that covers S by letting $P = \{p_c \mid c \in C\} \cup \{S\}$ and we write $\mathcal{S}_L^{\text{c}}$ ("c" for "channels") for the resulting LTS. One can also mix the two approaches and observe both locations and channels, giving rise to LTS's denoted $\mathcal{S}_L^{\text{l,c}}$.

Finally, we write $\mathcal{S}_L^{\mathrm{l,i}}$ ("i" for "idling") and, respectively, $\mathcal{S}_L^{\mathrm{c,i}}$, or $\mathcal{S}_L^{\mathrm{l,c,i}}$, for the variant STS's obtained by considering idling steps in the transition relation. For a nonempty $\alpha \subseteq \{\mathrm{i}, \mathrm{c}, \mathrm{l}\}$, we write LCS^α for the class of all $\mathcal{S}_L^\alpha$.

Observe that all variants of Lossy Channel Systems are WSTS's with strong compatibility when equipped with $\sqsubseteq$ as a wqo between states. Therefore they are in STS5 by Theorem 3.2.

In the next theorem we give tight results for all variants of lossy channel systems. When idling is allowed, LCS's are in STS4, otherwise they are in STS5, whatever the observable properties.

Theorem 4.4 • *The class* $\mathsf{LCS}^{\mathrm{l}}$ *gives rise to LTS's that are in STS*5 *but not in STS*4 *in general.*

- *The class* $\mathsf{LCS}^{\mathrm{i,l}}$ *gives rise to LTS's that are in STS*4 *but not in STS*3 *in general.*

Proof.

$\mathsf{LCS}^{\mathrm{l}}$**:** Let us give a counter-example to show that LCS's with locations as observable properties are not in STS4 in general. Consider the simple LCS L_1 in the left of Fig. 4 with only one rule $l \xrightarrow{?a} l$ (the name of the single channel is irrelevant). Starting from a configuration with n a's in the channel, a

Fig. 4. Two simple LCS's

trace of length n is possible but no longer trace is. As a consequence, trace equivalence does not have finite index and $\mathcal{S}_{L_1}^{\mathrm{l}}$ is not in STS4.

$\mathsf{LCS}^{\mathrm{i,l}}$**:** We first show that LCS's with idling are in STS4. To see this we consider the $\approx$ relation defined in the proof of Theorem 3.2: in the case of $\mathsf{LCS}^{\mathrm{i,l}}$, the proof that $\approx$ is a bounded-reach equivalence can be continued and, using idling steps, one shows that it is a distance equivalence.

For showing that in general $\mathsf{LCS}^{\mathrm{i,l}}$ does not give rise to systems in STS3, we consider the LCS L_2 in the right of Fig. 4. Starting from $\langle l, a^n \rangle$ there is a trace $p_l, p_{l'}, p_l, p_{l'}, \cdots$ of length n but no such trace longer than n (that is, longer traces must use idling steps and cannot alternate between p_l and $p_{l'}$). Hence trace equivalence does not have finite index and $\mathcal{S}_{L_2}^{\mathrm{i,l}}$ is not in STS3.

$\square$

Theorem 4.5 • *The class* $\mathsf{LCS}^{\mathrm{c}}$ *gives rise to LTS's that are in STS*5 *but not in STS*4 *in general.*

- *The class* $\mathsf{LCS}^{\mathrm{i,c}}$ *gives rise to LTS's that are in STS*4 *but not in STS*3 *in general.*

- *The class* $\mathsf{LCS}^{\mathrm{l,c}}$ *gives rise to LTS's that are in STS*5 *but not in STS*4 *in general.*
- *The class* $\mathsf{LCS}^{\mathrm{i,l,c}}$ *gives rise to LTS's that are in STS*4 *but not in STS*3 *in general.*

The proofs for these assertions (both positive parts and counter-examples) are very similar to the proof of Theorem 4.4 and are left to the reader.

5 Concluding remarks

We considered the STS hierarchy as a potential classification tool for various families of infinite-state models of systems. Given a class SC of systems (with its operational semantics), it is natural to ask the question of where the systems in SC fit in the STS hierarchy. This is a semantical question that can be answered independently of whether some region algebra and the associated algorithmics are available for class SC.

All previously known examples for levels STS1 to STS5 were some classes of hybrid or timed systems [HMR05]. We considered classical families of systems outside the world of timed/hybrid systems (Petri nets, pushdown systems, lossy channel systems) that support verification techniques. It turns out that only well-structured systems can fit in the STS hierarchy, and at the weakest levels (i.e., STS4 and STS5). As a side effect, we clarified the links between level STS5 and the well-structured systems of [FS01].

We are left with the conclusion that, at the moment, the STS hierarchy does not appear very enlightening outside the world of timed/hybrid systems or well-structured systems.

Acknowledgments

We thanks the anonymous referees for their many useful remarks and suggestions.

References

[AČJT00] P. A. Abdulla, K. Čerāns, B. Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *Information and Computation*, 160(1/2):109–127, 2000.

[AJ96] P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

[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.

[BEF+00] A. Bouajjani, J. Esparza, A. Finkel, O. Maler, P. Rossmanith, B. Willems, and P. Wolper. An efficient automata approach to some problems on context-free grammars. *Information Processing Letters*, 74(5–6):221–227, 2000.

[BS03] N. Bertrand and Ph. Schnoebelen. Model checking lossy channels systems is probably decidable. In *Proc. 6th Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS 2003), Warsaw, Poland, Apr. 2003*, volume 2620 of *Lecture Notes in Computer Science*, pages 120–135. Springer, 2003.

[Cau92] D. Caucal. On the regular structure of prefix rewriting. *Theoretical Computer Science*, 106(1):61–86, 1992.

[CFP96] G. Cécé, A. Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, 1996.

[Dic13] L. E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with r distinct prime factors. *Amer. Journal Math.*, 35:413–422, 1913.

[Esp98] J. Esparza. Decidability and complexity of Petri net problems — an introduction. In *Advances in Petri Nets 1998*, volume 1491 of *Lecture Notes in Computer Science*, pages 374–428. Springer, 1998.

[Fin87] A. Finkel. A generalization of the procedure of Karp and Miller to well structured transition systems. In *Proc. 14th Int. Coll. Automata, Languages, and Programming (ICALP '87), Karlsruhe, FRG, July 1987*, volume 267 of *Lecture Notes in Computer Science*, pages 499–508. Springer, 1987.

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

[FS01] A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.

[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.

[Hig52] G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc. (3)*, 2(7):326–336, 1952.

[HMR05] T. A. Henzinger, R. Majumdar, and J.-F. Raskin. A classification of symbolic transition systems. *ACM Trans. Computational Logic*, 6(1):1–32, 2005.