

# A General Approach to Comparing Infinite-State Systems with Their Finite-State Specifications

Antonín Kučera<sup>1\*</sup> and Philippe Schnoebelen<sup>2</sup>

<sup>1</sup> Faculty of Informatics, Masaryk University,  
Botanická 68a, 60200 Brno, Czech Republic,  
<tony@fi.muni.cz>

<sup>2</sup> LSV, ENS de Cachan & CNRS UMR 8643,  
61, av. Pdt. Wilson, 94235 Cachan Cedex, France.  
<phs@lsv.ens-cachan.fr>

**Abstract.** We introduce a generic family of behavioral relations for which the problem of comparing an arbitrary transition system to some finite-state specification can be reduced to a model checking problem against simple modal formulae. As an application, we derive decidability of several regular equivalence problems for well-known families of infinite-state systems.

## 1 Introduction

Verification of infinite-state models of systems is a very active field of research, see [9, 8, 5, 19, 31] for surveys of some subfields. In this area, researchers consider a large variety of models suited to different kinds of applications, and three main kinds of verification problems: (1) specific properties like reachability or termination, (2) model checking of temporal formulae, and (3) semantic equivalences or preorders between two systems. With most models, termination and reachability are investigated first. Positive results lead to investigations of more general temporal model checking problems. Regarding equivalence problems, positive decidability results exist mainly for strong bisimilarity (some milestones in the study include [3, 13, 12, 14, 11, 30]). For other behavioral equivalences, results are usually negative.

*Regular equivalence problem.* Recently, the problem of comparing some infinite-state process  $g$  with a *finite-state* specification  $f$  has been identified as an important subcase<sup>3</sup> of the general equivalence checking problem [19]. Indeed, in equivalence-based verification, one usually compares a “real-life” system with an abstract behavioral specification. Faithful models of real-life systems often require features like counters, sub-process creation, or unbounded buffers, that make the model infinite-state. On the other

---

\* On leave at LSV, ENS de Cachan, France. Supported by the Grant Agency of the Czech Republic, grant No. 201/03/1161.

<sup>3</sup> We refer to this subcase as “the regular equivalence problem” in the rest of this paper. For example, if we say that “regular weak bisimilarity is decidable for PA processes”, we mean that weak bisimilarity is decidable between PA processes and finite-state ones.

hand, the behavioral specification is usually abstract, hence naturally finite-state. Moreover, infinite-state systems are often abstracted to finite-state systems even before applying further analytical methods. This approach naturally subsumes the question if the constructed abstraction is correct (i.e., equivalent to the original system). It quickly appeared that regular equivalence problems are computationally easier than comparing two infinite-state processes, and a wealth of positive results exist [19].

The literature offers two generic techniques for deciding regular equivalences. First, Abdulla *et al.* show how to check *regular simulation* on *well-structured* processes [2]. Their algorithm is generic because a large collection of infinite-state models are well-structured [10].

The second approach is even more general: one expresses equivalence with  $f$  via a formula  $\varphi_f$  of some modal logic  $\mathcal{L}$ .  $\varphi_f$  is called a *characteristic formula* for  $f$  wrt. the given equivalence. This reduces regular equivalence problems to more familiar model checking problems. It entails decidability of regular equivalences for all systems where model checking with the logic  $\mathcal{L}$  is decidable. It is easy to give characteristic formulae wrt. bisimulation-like equivalences if one uses the modal  $\mu$ -calculus [32, 26]. Browne *et al.* constructed characteristic formulae wrt. bisimilarity and branching-bisimilarity in the logic CTL [7]. Unfortunately, CTL (or  $\mu$ -calculus) model checking is undecidable on many process classes like PA, Petri nets, lossy channel systems, etc. Later, it has been shown that characteristic formulae wrt. strong and weak bisimilarity can be constructed even in the  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  fragment of CTL [15]. This logic is sufficiently simple and its associated model-checking problem is decidable in many classes of infinite-state systems (including PA, lossy channel systems, and pushdown automata) [24].

*Our contribution.* We introduce *full regular equivalences*, a variant of regular equivalences, and develop a generic approach to the reduction of full regular equivalences to model checking (essentially) the EF fragment of modal logic<sup>4</sup>. Compared to regular equivalences, full regular equivalence has the additional requirement that the state-space of the infinite system must be included in the state-space of the finite system up to the given equivalence. We argue that full regular equivalence is as natural as regular equivalence in most practical situations (additionally the two variants turn out to coincide in many cases). Moreover, an important outcome of our results is that full regular equivalence is “more decidable” than regular equivalence for trace-like and simulation-like equivalences. For example, regular trace equivalence is undecidable for BPA (and hence also for pushdown and PA processes), while full regular trace equivalence is decidable for these models. Similar examples can be given for simulation-like equivalences. See Section 2 and Section 6 for further comments.

We offer two main reductions. One applies to a large parameterized family of equivalences defined via a transfer property (we call them *MTB* equivalences). The other applies to a large parameterized family of equivalences based on sets of enriched traces (we call them *PQ* equivalences). Together they cover virtually all process equivalences used in verification [33]. For all of these, full regular equivalence with some  $f$  is reduced to EF model-checking, hence shown decidable for a large family of infinite-state mod-

<sup>4</sup> In fact we provide reductions to  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  and to  $\mathcal{L}(\mathbf{EU}_\alpha, \mathbf{EF})$ , two different fragments of modal logic that have incomparable expressive power.

els. More precisely, the constructions output a *characteristic formula* for  $f$  wrt. a given equivalence, which expresses the property of “being fully equivalent to  $f$ ”. In particular, this works for bisimulation-like equivalences (weak, delay, early, branching), and thus we also obtain a refinement of the result presented in [7] which says that a characteristic formula wrt. branching bisimilarity is constructible in CTL. The main “message” of this part is that full regular equivalence is decidable for many more semantic equivalences and classes of infinite-state models than regular equivalence. In this paper we do not aim to develop specific methods for particular models and equivalences. (Such methods can be more efficient than our generic (model-independent) algorithm—for example, it has recently been shown in [20] that full regular equivalence with PDA processes can be decided by a PDA-specific algorithm which needs only polynomial time for some *MTB* equivalences and some subclasses of PDA processes.)

Another contribution of this paper is a model-checking algorithm for the logic  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha)$  and lossy channel systems. This allows one to apply the previous abstract results also to processes of lossy channel systems (for other models like, e.g., pushdown automata, PA processes, or PAD processes, the decidability of model-checking problem with the logic EF is already known).

Due to space constraints, we had to omit all proofs. These can be found in a full version of this paper [21].

## 2 (Full) Regular Equivalence

We start by recalling basic definitions. Let  $Act = \{a, b, c, \dots\}$  be a countably infinite set of *actions*, and let  $\tau \notin Act$  be a distinguished *silent action*. For  $\mathcal{A} \subseteq Act$ ,  $\mathcal{A}_\tau$  denotes the set  $\mathcal{A} \cup \{\tau\}$ . We use  $\alpha, \beta, \dots$  to range over  $Act_\tau$ . A *transition system* is a triple  $\mathcal{T} = (S, \rightarrow, \mathcal{A})$  where  $S$  is a set of *states*,  $\mathcal{A} \subseteq Act_\tau$  is a finite *alphabet*, and  $\rightarrow \subseteq S \times \mathcal{A} \times S$  is a *transition relation*. We write  $s \xrightarrow{\alpha} t$  instead of  $(s, \alpha, t) \in \rightarrow$ , and we extend this notation to elements of  $\mathcal{A}^*$  in the standard way. We say that a state  $t$  is *reachable* from a state  $s$ , written  $s \rightarrow^* t$ , if there is  $w \in \mathcal{A}^*$  such that  $s \xrightarrow{w} t$ . Further, for every  $\alpha \in Act_\tau$  we define the relation  $\xrightarrow{\alpha} \subseteq S \times S$  as follows:  $s \xrightarrow{\tau} t$  iff there is a sequence of the form  $s = p_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} p_k = t$  where  $k \geq 0$ ;  $s \xrightarrow{a} t$  where  $a \neq \tau$  iff there are  $p, q$  such that  $s \xrightarrow{\tau} p \xrightarrow{a} q \xrightarrow{\tau} t$ . From now on, a *process* is formally understood as a state of (some) transition system. Intuitively, transitions from a given process  $s$  model possible computational steps, and the silent action  $\tau$  is used to mark those steps which are internal (i.e., not externally observable). Since we sometimes consider processes without explicitly defining their associated transition systems, we also use  $\mathcal{A}(s)$  to denote the alphabet of (the underlying transition system of) the process  $s$ . A process  $s$  is  $\tau$ -*free* if  $\tau \notin \mathcal{A}(s)$ .

Let  $\sim$  be an arbitrary process equivalence,  $g$  a (general) process,  $\mathcal{F}$  a finite-state system, and  $f$  a process of  $\mathcal{F}$ .

**Definition 1 (Full Regular Equivalence).** *We say  $g$  is fully equivalent to  $f$  (in  $\mathcal{F}$ ) iff:*

- $g \sim f$  ( $g$  is equivalent to  $f$ ), and
- for all  $g \rightarrow^* g'$ , there is some  $f'$  in  $\mathcal{F}$  s.t.  $g' \sim f'$  (every process reachable from  $g$  has an equivalent in  $\mathcal{F}$ ).

Observe that the equivalent  $f'$  does *not* have to be reachable from  $f$ .

In verification settings, requiring that some process  $g$  is fully equivalent to a finite-state specification  $\mathcal{F}$  puts some additional constraints on  $g$ : its whole state-space must be accounted for in a finite way. To get some intuition why this is meaningful, consider, e.g., the finite-state system with three states  $f, f', f''$  and transitions  $f \xrightarrow{a} f, f' \xrightarrow{a} f''$ . Suppose that all transitions of a given infinite-state system  $g$  are labeled by  $a$ . Then regular trace equivalence to  $f$  means that  $g$  can do infinitely many  $a$ 's (assuming that  $g$  is finitely branching), while full regular trace equivalence to  $f$  means that  $g$  can do infinitely many  $a$ 's and whenever it decides to terminate, it can reach a terminated state in at most one transition. This property cannot be encoded as regular bisimulation equivalence or regular simulation equivalence by any finite-state system. Let us also note that when  $\sim$  is an equivalence of the bisimulation family, then regular equivalence is automatically “full”.

### 3 MTB Preorder and Equivalence

In this paper, we aim to prove general results about equivalence-checking between infinite-state and finite-state processes. To achieve that, we consider an abstract notion of process preorder and process equivalence which will be introduced next.

A *transfer* is one of the three operators on binary relations defined as follows:  $\text{sim}(R) = R$ ,  $\text{bisim}(R) = R \cap R^{-1}$ ,  $\text{contrasim}(R) = R^{-1}$ . A *mode* is a subset of  $\{\eta, d\}$  (the  $\eta$  and  $d$  are just two different symbols). A *basis* is an equivalence over processes satisfying the following property: whenever  $(s, u) \in B$  and  $s \xrightarrow{\tau} t \xrightarrow{\tau} u$ , then also  $(s, t) \in B$ .

**Definition 2.** Let  $\mathcal{S}$  be a binary relation over processes and  $M$  a mode. A move  $s \xrightarrow{\alpha} t$  is tightly  $\mathcal{S}$ -consistent with  $M$  if either  $\alpha = \tau$  and  $s = t$ , or there is a sequence  $s = s_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} s_k \xrightarrow{\alpha} t_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} t_\ell = t$ , where  $k, \ell \geq 0$ , such that the following holds: (1) if  $\eta \in M$ , then  $(s_i, s_j) \in \mathcal{S}$  for all  $0 \leq i, j \leq k$ ; (2) if  $d \in M$ , then  $(t_i, t_j) \in \mathcal{S}$  for all  $0 \leq i, j \leq \ell$ .

The loose  $\mathcal{S}$ -consistency of  $s \xrightarrow{\alpha} t$  with  $M$  is defined in the same way, but the conditions (1), (2) are weakened—we only require that  $(s_0, s_k), (s_k, s_0) \in \mathcal{S}$ , and  $(t_0, t_\ell), (t_\ell, t_0) \in \mathcal{S}$ .

**Definition 3.** Let  $T$  be a transfer,  $M$  a mode, and  $B$  a basis. A binary relation  $\mathcal{R}$  over processes is a tight (or loose) MTB-relation if it satisfies the following:

- $\mathcal{R} \subseteq B$
- whenever  $(p, q) \in \mathcal{R}$ , then for every tightly (or loosely, resp.)  $\mathcal{R}$ -consistent move  $p \xrightarrow{\alpha} p'$  there is a tightly (or loosely, resp.)  $\mathcal{R}$ -consistent move  $q \xrightarrow{\alpha} q'$  such that  $(p', q') \in T(\mathcal{R})$ .

We write  $s \sqsubseteq t$  (or  $s \preceq t$ , resp.), if there is a tight (or loose, resp.) MTB-relation  $\mathcal{R}$  such that  $(s, t) \in \mathcal{R}$ . We say that  $s, t$  are tightly (or loosely, resp.) MTB-equivalent, written  $s \sim t$  (or  $s \approx t$ , resp.), if  $s \sqsubseteq t$  and  $t \sqsubseteq s$  (or  $s \preceq t$  and  $t \preceq s$ , resp.).

It is standard that such a definition entails that  $\sqsubseteq$  and  $\preceq$  are preorders, and  $\sim$  and  $\approx$  are equivalences over the class of all processes. The relationship between  $\sqsubseteq$  and  $\preceq$  relations is clarified in the next lemma (this is where we need the defining property of a base).

**Lemma 4.** *We have that  $\sqsubseteq = \preceq$  (and hence also  $\sim = \approx$ ).*

Before presenting further technical results, let us briefly discuss and justify the notion of *MTB* equivalence. The class of all *MTB* equivalences can be partitioned into the subclasses of simulation-like, bisimulation-like, and contrasimulation-like equivalences according to the chosen transfer. Additional conditions which must be satisfied by equivalent processes can be specified by an appropriately defined base. For example, we can put  $B$  to be *true*, *ready*, or *terminate* where

- $(s, t) \in \textit{true}$  for all  $s$  and  $t$ ;
- $(s, t) \in \textit{ready}$  iff  $\{a \in \textit{Act}_\tau \mid \exists s' : s \xrightarrow{a} s'\} = \{a \in \textit{Act}_\tau \mid \exists t' : t \xrightarrow{a} t'\}$ ;
- $(s, t) \in \textit{terminate}$  iff  $s$  and  $t$  are either both terminating, or both non-terminating (a process  $p$  is terminating iff  $p \xrightarrow{\alpha} p'$  implies  $\alpha = \tau$  and  $p$  cannot perform an infinite sequence of  $\tau$ -transitions).

The mode specifies the level of ‘control’ over the states that are passed through by  $\xrightarrow{\alpha}$  transitions. In particular, by putting  $T = \textit{bisim}$ ,  $B = \textit{true}$ , and choosing  $M$  to be  $\emptyset$ ,  $\{\eta\}$ ,  $\{d\}$ , or  $\{\eta, d\}$ , one obtains weak bisimilarity [25],  $\eta$ -bisimilarity [4], delay-bisimilarity, and branching bisimilarity [34], respectively.<sup>5</sup> “Reasonable” refinements of these bisimulation equivalences can be obtained by redefining  $B$  to something like *terminate*—sometimes there is a need to distinguish between, e.g., terminated processes and processes which enter an infinite internal loop. If we put  $T = \textit{sim}$ ,  $B = \textit{true}$ , and  $M = \emptyset$ , we obtain weak simulation equivalence; and by redefining  $B$  to *ready* we yield a variant of ready simulation equivalence. The equivalence where  $T = \textit{contrasim}$ ,  $B = \textit{true}$ , and  $M = \emptyset$  is known as contrasimulation (see, e.g., [35]).<sup>6</sup>

The definition of *MTB* equivalence allows to combine all of the three parameters arbitrarily, and our results are valid for all such combinations (later we adopt some natural effectiveness assumptions about  $B$ , but this will be the only restriction).

**Definition 5.** *For every  $k \in \mathbb{N}_0$ , the binary relations  $\sqsubseteq_k$ ,  $\sim_k$ ,  $\preceq_k$ , and  $\approx_k$  are defined as follows:  $s \sqsubseteq_0 t$  iff  $(s, t) \in B$ ;  $s \sqsubseteq_{k+1} t$  iff  $(s, t) \in B$  and for every tightly  $\sqsubseteq_k$ -consistent move  $s \xrightarrow{\alpha} s'$  there is some tightly  $\sqsubseteq_k$ -consistent move  $t \xrightarrow{\alpha} t'$  such that  $(s', t') \in T(\sqsubseteq_k)$ .*

<sup>5</sup> Our definition of *MTB* equivalence does not directly match the definitions of  $\eta$ -, delay-, and branching bisimilarity that one finds in the literature. However, it is easy to show that one indeed yields exactly these equivalences.

<sup>6</sup> Contrasimulation can also be seen as a generalization of coupled simulation [27, 28], which was defined only for the subclass of divergence-free processes (where it coincides with contrasimulation). It is worth to note that contrasimulation coincides with strong bisimilarity on the subclass of  $\tau$ -free processes (to see this, realize that one has to consider the moves  $s \xrightarrow{\tau} s$  even if  $s$  is  $\tau$ -free). This is (intuitively) the reason why contrasimulation has some nice properties also in the presence of silent moves.

The  $\preceq_k$  relations are defined in the same way, but we require only loose  $\preceq_k$ -consistency of moves in the inductive step. Finally, we put  $s \sim_k t$  iff  $s \sqsubseteq_k t$  and  $t \sqsubseteq_k s$ , and similarly  $s \approx_k t$  iff  $s \preceq_k t$  and  $t \preceq_k s$ .

A trivial observation is that  $\preceq_k \supseteq \preceq_{k+1} \supseteq \preceq$ ,  $\sqsubseteq_k \supseteq \sqsubseteq_{k+1} \supseteq \sqsubseteq$ ,  $\sim_k \supseteq \sim_{k+1} \supseteq \sim$ , and  $\approx_k \supseteq \approx_{k+1} \supseteq \approx$  for each  $k \in \mathbb{N}_0$ . In general,  $\sqsubseteq_k \neq \preceq_k$ ; however, if we restrict ourselves to processes of some fixed finite-state system, we can prove the following:

**Lemma 6.** *Let  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  be a finite-state system with  $n$  states. Then  $\sqsubseteq_{n^2-1} = \sqsubseteq_{n^2} = \sqsubseteq = \preceq = \preceq_{n^2-1} = \preceq_{n^2}$ , where all of the relations are considered as being restricted to  $F \times F$ .*

**Theorem 7.** *Let  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  be a finite-state system with  $n$  states,  $f$  a process of  $F$ , and  $g$  some (arbitrary) process. Then the following three conditions are equivalent.*

- (a)  $g \sim f$  and for every  $g \rightarrow^* g'$  there is some  $f' \in F$  such that  $g' \sim f'$ .
- (b)  $g \sim_{n^2} f$  and for every  $g \rightarrow^* g'$  there is some  $f' \in F$  such that  $g' \sim_{n^2} f'$ .
- (c)  $g \approx_{n^2} f$  and for every  $g \rightarrow^* g'$  there is some  $f' \in F$  such that  $g' \approx_{n^2} f'$ .

### 3.1 Encoding MTB Equivalence into Modal Logic

In this section we show that the conditions (b) and (c) of Theorem 7 can be expressed in modal logic. Let us consider a class of modal formulae defined by the following abstract syntax equation (where  $\alpha$  ranges over  $Act_\tau$ ):

$$\varphi ::= \text{tt} \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \mathbf{EX}_\alpha \varphi \mid \mathbf{EF} \varphi \mid \mathbf{EF}_\tau \varphi \mid \varphi_1 \mathbf{EU}_\alpha \varphi_2$$

The semantics (over processes) is defined inductively as follows:

- $s \models \text{tt}$  for every process  $s$ .
- $s \models \varphi_1 \wedge \varphi_2$  iff  $s \models \varphi_1$  and  $s \models \varphi_2$ .
- $s \models \neg\varphi$  iff  $s \not\models \varphi$ .
- $s \models \mathbf{EX}_\alpha \varphi$  iff there is  $s \xrightarrow{\alpha} s'$  such that  $s' \models \varphi$ .
- $s \models \mathbf{EF} \varphi$  iff there is  $s \rightarrow^* s'$  such that  $s' \models \varphi$ .
- $s \models \mathbf{EF}_\tau \varphi$  iff there is  $s \xrightarrow{\tau} s'$  such that  $s' \models \varphi$ .
- $s \models \varphi_1 \mathbf{EU}_\alpha \varphi_2$  iff either  $\alpha = \tau$  and  $s \models \varphi_2$ , or there is a sequence  $s = s_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} s_m \xrightarrow{\alpha} s'$ , where  $m \geq 0$ , such that  $s_i \models \varphi_1$  for all  $0 \leq i \leq m$  and  $s' \models \varphi_2$ .

The dual operator to  $\mathbf{EF}$  is  $\mathbf{AG}$ , defined by  $\mathbf{AG} \varphi \equiv \neg \mathbf{EF} \neg\varphi$ .

Let  $M_1, \dots, M_k$  range over  $\{\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha\}$ . The (syntax of the) logic  $\mathcal{L}(M_1, \dots, M_k)$  consists of all modal formulae built over the modalities  $M_1, \dots, M_k$ .

Let  $\sim$  be an MTB equivalence. Our aim is to show that for every finite  $f$  there are formulae  $\varphi_f$  of  $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$  and  $\psi_f$  of  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  such that for every process  $g$  where  $\mathcal{A}(g) \subseteq \mathcal{A}$  we have that  $g \models \varphi_f$  (or  $g \models \psi_f$ ) iff the processes  $g$  and  $f$  satisfy the condition (b) (or (c), resp.) of Theorem 7. Clearly such formulae cannot always exist without some additional assumptions about the base  $B$ . Actually, all we need is to assume that the equivalence  $B$  with processes of a given finite-state system  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  is definable in the aforementioned logics. More precisely, for

each  $f \in F$  there should be formulae  $\Xi_f^t$  and  $\Xi_f^\ell$  of the logics  $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$  and  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ , respectively, such that for every process  $g$  where  $\mathcal{A}(g) \subseteq \mathcal{A}$  we have that  $(g, f) \in B$  iff  $g \models \Xi_f^t$  iff  $g \models \Xi_f^\ell$ . Since we are also interested in complexity issues, we further assume that the formulae  $\Xi_f^t$  and  $\Xi_f^\ell$  are *efficiently* computable from  $\mathcal{F}$ . An immediate consequence of this assumption is that  $B$  over  $F \times F$  is efficiently computable. This is because the model-checking problem with  $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$  and  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  is decidable in polynomial time over finite-state systems. To simplify the presentation of our complexity results, we adopt the following definition:

**Definition 8.** *We say that a base  $B$  is well-defined if there is a polynomial  $\mathcal{P}$  (in two variables) such that for every finite-state system  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  the set  $\{\Xi_f^t, \Xi_f^\ell \mid f \in F\}$  can be computed, and the relation  $B \cap (F \times F)$  can be decided, in time  $\mathcal{O}(\mathcal{P}(|F|, |\mathcal{A}|))$ .*

*Remark 9.* Note that a well-defined  $B$  is not necessarily decidable over process classes which contain infinite-state processes—for example, the *ready* base introduced in the previous section is well-defined but it is not decidable for, e.g., CCS processes. In fact, the  $\Xi_f^t$  formulae are only required for the construction of  $\varphi_f$ , and the  $\Xi_f^\ell$  formulae are required only for the construction of  $\psi_f$ . (This is why we provide two different formulae for each  $f$ .) Note that there are bases for which we can construct only one of the  $\Xi_f^t$  and  $\Xi_f^\ell$  families, which means that for some *MTB* equivalences we can construct only one of the  $\varphi_f$  and  $\psi_f$  formulae. A concrete example is the *terminate* base of the previous section, which is definable in  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  but not in  $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$ .  $\square$

For the rest of this section, we fix some *MTB*-equivalence  $\sim$  where  $B$  is well-defined, and a finite-state system  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  with  $n$  states.

Let  $\langle \alpha, \varphi_\eta, \varphi_d \rangle^t$  and  $\langle \alpha, \varphi_\eta, \varphi_d \rangle^\ell$  be unary modal operators whose semantics is defined as follows:

- $s \models \langle \alpha, \varphi_\eta, \varphi_d \rangle^t \varphi$  iff either  $\alpha = \tau$  and  $s \models \varphi$ , or there is a sequence of the form  $s = p_0 \xrightarrow{\tau} \dots p_k \xrightarrow{\alpha} q_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} q_m$ , where  $k, m \geq 0$ , such that  $p_i \models \varphi_\eta$  for all  $0 \leq i \leq k$ ,  $q_j \models \varphi_d$  for all  $0 \leq j \leq m$ , and  $q_m \models \varphi$ .
- $s \models \langle \alpha, \varphi_\eta, \varphi_d \rangle^\ell \varphi$  iff either  $\alpha = \tau$  and  $s \models \varphi$ , or there is a sequence of the form  $s = p_0 \xrightarrow{\tau} \dots p_k \xrightarrow{\alpha} q_0 \xrightarrow{\tau} \dots \xrightarrow{\tau} q_m$ , where  $k, m \geq 0$ , such that  $p_0 \models \varphi_\eta$ ,  $p_k \models \varphi_\eta$ ,  $q_0 \models \varphi_d$ ,  $q_m \models \varphi_d$ , and  $q_m \models \varphi$ .

We also define  $[\alpha, \varphi_\eta, \varphi_d]^t \varphi$  as an abbreviation for  $\neg \langle \alpha, \varphi_\eta, \varphi_d \rangle^t \neg \varphi$ , and similarly  $[\alpha, \varphi_\eta, \varphi_d]^\ell \varphi$  is used to abbreviate  $\neg \langle \alpha, \varphi_\eta, \varphi_d \rangle^\ell \neg \varphi$ .

**Lemma 10.** *The  $\langle \alpha, \varphi_\eta, \varphi_d \rangle^t$  and  $\langle \alpha, \varphi_\eta, \varphi_d \rangle^\ell$  modalities are expressible in  $\mathcal{L}(\mathbf{EU}_\alpha)$  and  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ , respectively:*

Since the conditions (b) and (c) of Theorem 7 are encoded into  $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$  and  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  along the same scheme, we present both constructions at once by adopting the following notation:  $\langle \alpha, \varphi_\eta, \varphi_d \rangle$  stands either for  $\langle \alpha, \varphi_\eta, \varphi_d \rangle^t$  or  $\langle \alpha, \varphi_\eta, \varphi_d \rangle^\ell$ ,  $\Xi_f$  denotes either  $\Xi_f^t$  or  $\Xi_f^\ell$ ,  $\overset{\circ}{\simeq}_k$  denotes either  $\sim_k$  or  $\approx_k$ , and  $\leq_k$  denotes either  $\sqsubseteq_k$  or  $\preceq_k$ , respectively. Moreover, we write  $s \xrightarrow{\alpha, k} t$  to denote that there is either a tightly  $\sqsubseteq_k$ -consistent move  $s \xrightarrow{\alpha} t$ , or a loosely  $\preceq_k$ -consistent move  $s \xrightarrow{\alpha} t$ , respectively.

**Definition 11.** For all  $f \in F$  and  $k \in \mathbb{N}_0$  we define the formulae  $\Phi_{f,k}$ ,  $\Psi_{f,k}$ , and  $\Theta_{f,k}$  inductively as follows:

- $\Phi_{f,0} = \Psi_{f,0} = \Xi_f$
- $\Theta_{f,k} = \Phi_{f,k} \wedge \Psi_{f,k}$
- $\Phi_{f,k+1} = \Xi_f \wedge (\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}) \wedge (\bigwedge_{f \xrightarrow{\alpha,k} f'} (\bigvee_{f_1, f_2 \in F} (\alpha, \varphi_{f_1,k}, \psi_{f_2,k}) \xi_{f',k}))$
- $\Psi_{f,k+1} = \Xi_f \wedge (\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}) \wedge \bigwedge_{\alpha \in \mathcal{A}_\tau, f_1, f_2 \in F} ([\alpha, \varphi_{f_1,k}, \psi_{f_2,k}] (\bigvee_{f \xrightarrow{\alpha,k} f'} \varrho_{f',k}))$

where

- if  $\eta \in M$ , then  $\varphi_{f_1,k} = \Theta_{f_1,k}$ , otherwise  $\varphi_{f_1,k} = \mathbf{tt}$ ;
- if  $d \in M$ , then  $\psi_{f_2,k} = \Theta_{f_2,k}$ , otherwise  $\psi_{f_2,k} = \mathbf{tt}$ ;
- if  $T = \text{sim}$ , then  $\xi_{f',k} = \Phi_{f',k}$  and  $\varrho_{f',k} = \Psi_{f',k}$ ;
- if  $T = \text{bisim}$ , then  $\xi_{f',k} = \varrho_{f',k} = \Theta_{f',k}$ ;
- if  $T = \text{contrasim}$ , then  $\xi_{f',k} = \Psi_{f',k}$  and  $\varrho_{f',k} = \Phi_{f',k}$ .

The empty conjunction is equivalent to  $\mathbf{tt}$ , and the empty disjunction to  $\mathbf{ff}$ .

The meaning of the constructed formulae is explained in the next theorem. Intuitively, what we *would like* to have is that for every process  $g$  where  $\mathcal{A}(g) \subseteq \mathcal{A}$  it holds that  $g \models \Phi_{f,k}$  iff  $f \leq_k g$ , and  $g \models \Psi_{f,k}$  iff  $g \leq_k f$ . However, this is (provably) *not achievable*—the  $\leq_k$  preorder with a given finite-state process is not directly expressible in the logics  $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$  and  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ . The main trick (and subtlety) of the presented inductive construction is that the formulae  $\Phi_{f,k}$  and  $\Psi_{f,k}$  actually express *stronger* conditions.

**Theorem 12.** Let  $g$  be an (arbitrary) process such that  $\mathcal{A}(g) \subseteq \mathcal{A}$ . Then for all  $f \in F$  and  $k \in \mathbb{N}_0$  we have the following:

- (a)  $g \models \Phi_{f,0}$  iff  $f \leq_0 g$ ; further,  $g \models \Phi_{f,k+1}$  iff  $f \leq_{k+1} g$  and for each  $g \xrightarrow{*} g'$  there is  $f' \in F$  such that  $g' \stackrel{\circ}{\leq}_k f'$ .
- (b)  $g \models \Psi_{f,0}$  iff  $g \leq_0 f$ ; further,  $g \models \Psi_{f,k+1}$  iff  $g \leq_{k+1} f$  and for each  $g \xrightarrow{*} g'$  there is  $f' \in F$  such that  $g' \stackrel{\circ}{\leq}_k f'$ .
- (c)  $g \models \Theta_{f,0}$  iff  $g \stackrel{\circ}{\leq}_0 f$ ; further,  $g \models \Theta_{f,k+1}$  iff  $f \stackrel{\circ}{\leq}_{k+1} g$  and for each  $g \xrightarrow{*} g'$  there is  $f' \in F$  such that  $g' \stackrel{\circ}{\leq}_k f'$ .

In general, the  $\leq_k$ -consistency of moves  $g \xrightarrow{\alpha} g'$  can be expressed in a given logic only if one can express the  $\stackrel{\circ}{\leq}_k$  equivalence with  $g$  and  $g'$ . Since  $g$  and  $g'$  can be infinite-state processes, this is generally impossible. This difficulty was overcome in Theorem 12 by using the assumption that  $g$  and  $g'$  are  $\stackrel{\circ}{\leq}_k$  equivalent to some  $f_1$  and  $f_2$  of  $F$ . Thus, we only needed to encode the  $\stackrel{\circ}{\leq}_k$  equivalence with  $f_1$  and  $f_2$  which is (in a way) achieved by the  $\Theta_{f_1,k}$  and  $\Theta_{f_2,k}$  formulae. An immediate consequence of Theorem 7 and Theorem 12 is the following:

**Corollary 13.** Let  $g$  be an (arbitrary) process such that  $\mathcal{A}(g) \subseteq \mathcal{A}$ , and let  $f \in F$ . Then the following two conditions are equivalent:

- (a)  $g \sim f$  and for every  $g \xrightarrow{*} g'$  there is some  $f' \in F$  such that  $g' \sim f'$ .
- (b)  $g \models \Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n^2})$ .

Since the formula  $\Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n^2})$  is effectively constructible, the problem (a) of the previous corollary is effectively reducible to the problem (b). A natural question is what is the complexity of the reduction from (a) to (b). At first glance, it seems to be exponential because the size of  $\Theta_{f',n^2}$  is exponential in the size of  $\mathcal{F}$ . However, the number of distinct subformulae in  $\Theta_{f',n^2}$  is only *polynomial*. This means that if we represent the formula  $\Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n^2})$  by a *circuit*<sup>7</sup>, then the size of this circuit is only polynomial in the size of  $\mathcal{F}$ . This is important because the complexity of many model-checking algorithms actually depends on the size of the circuit representing a given formula rather than on the size of the formula itself. The size of the circuit for  $\Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n^2})$  is estimated in our next lemma.

**Lemma 14.** *The formula  $\Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n^2})$  can be represented by a circuit constructible in  $\mathcal{O}(n^6 \cdot |\mathcal{A}| + \mathcal{P}(n, |\mathcal{A}|))$  time.*

## 4 PQ Preorder and Equivalence

Let  $M, N$  be sets of processes. We write  $M \overset{\alpha}{\Rightarrow} N$  iff for every  $t \in N$  there is some  $s \in M$  such that  $s \overset{\alpha}{\Rightarrow} t$ . In the next definition we introduce another parametrized equivalence which is an abstract template for trace-like equivalences.

**Definition 15.** *Let  $P$  be a preorder over the class of all processes and let  $Q \in \{\forall, \exists\}$ . For every  $i \in \mathbb{N}_0$  we inductively define the relation  $\sqsubseteq_i$  as follows:*

- $s \sqsubseteq_0 M$  for every process  $s$  and every set of processes  $M$  such that
  - if  $Q = \forall$ , then  $(s, t) \in P$  for every  $t \in M$ ;
  - if  $Q = \exists$ , then  $(s, t) \in P$  for some  $t \in M$ ;
- $s \sqsubseteq_{i+1} M$  if  $s \sqsubseteq_i M$  and for every  $s \overset{\alpha}{\Rightarrow} t$  there is  $M \overset{\alpha}{\Rightarrow} N$  such that  $t \sqsubseteq_i N$ .

*Slightly abusing notation, we write  $s \sqsubseteq_i t$  instead of  $s \sqsubseteq_i \{t\}$ . Further, we define the PQ preorder, denoted “ $\sqsubseteq$ ”, by  $s \sqsubseteq M$  iff  $s \sqsubseteq_i M$  for every  $i \in \mathbb{N}_0$ . Processes  $s, t$  are PQ equivalent, written  $s \sim t$ , iff  $s \sqsubseteq t$  and  $t \sqsubseteq s$ .*

For every process  $s$ , let  $I(s) = \{a \in \text{Act} \mid s \overset{a}{\Rightarrow} t \text{ for some } t\}$  (note that  $\tau \notin I(s)$ ). Now consider the preorders  $T, D, F, R, S$  defined as follows:

- $(s, t) \in T$  for all  $s, t$  (true).
- $(s, t) \in D$  iff both  $I(s)$  and  $I(t)$  are either empty or non-empty (deadlock equivalence).
- $(s, t) \in F$  iff  $I(s) \supseteq I(t)$  (failure preorder).
- $(s, t) \in R$  iff  $I(s) = I(t)$  (ready equivalence).
- $(s, t) \in S$  iff  $s$  and  $t$  are trace equivalent (that is, iff  $\{w \in \text{Act}^* \mid \exists s \overset{w}{\Rightarrow} s'\} = \{w \in \text{Act}^* \mid \exists t \overset{w}{\Rightarrow} t'\}$ ).

Now one can readily check that  $TQ, D\exists, F\exists, F\forall, R\exists, R\forall$ , and  $S\exists$  equivalence is in fact trace, completed trace, failure, failure trace, readiness, ready trace, and possible futures equivalence, respectively. Other trace-like equivalences can be defined similarly.

<sup>7</sup> A circuit (or a DAG) representing a formula  $\varphi$  is basically the syntax tree for  $\varphi$  where the nodes representing the same subformula are identified.

**Lemma 16.** Let  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  be a finite-state system with  $n$  states. Then  $\sqsubseteq_{n2^{n-1}} = \sqsubseteq_{n2^n} = \sqsubseteq$ , where all of the relations are considered as being restricted to  $F \times 2^F$ .

**Lemma 17.** For all  $i \in \mathbb{N}_0$ , processes  $s, t$ , and sets of processes  $M, N$  we have that

- (a) if  $s \sqsubseteq_i t$  and  $t \sqsubseteq_i M$ , then also  $s \sqsubseteq_i M$ ;
- (b) if  $s \sqsubseteq_i M$  and for every  $u \in M$  there is some  $v \in N$  such that  $u \sqsubseteq_i v$ , then also  $s \sqsubseteq_i N$ .

**Theorem 18.** Let  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  be a finite-state system with  $n$  states,  $f$  a process of  $F$ , and  $g$  some (arbitrary) process. Then the following two conditions are equivalent.

- (a)  $g \sim f$  and for every  $g \rightarrow^* g'$  there is some  $f' \in F$  such that  $g' \sim f'$ .
- (b)  $g \sim_{n2^n} f$  and for every  $g \rightarrow^* g'$  there is some  $f' \in F$  such that  $g' \sim_{n2^n} f'$ .

Now we show how to encode the condition (b) of Theorem 18 into modal logic. To simplify our notation, we introduce the  $\langle\langle\alpha\rangle\rangle$  operator defined as follows:  $\langle\langle\alpha\rangle\rangle\varphi$  stands either for  $\mathbf{EF}_\tau\varphi$  (if  $\alpha = \tau$ ), or  $\mathbf{EF}_\tau\mathbf{EX}_\alpha\mathbf{EF}_\tau\varphi$  (if  $\alpha \neq \tau$ ). Moreover,  $\llbracket\alpha\rrbracket\varphi \equiv \neg\langle\langle\alpha\rangle\rangle\neg\varphi$ . Similarly as in the case of *MTB* equivalence, we need some effectiveness assumptions about the preorder  $P$ , which are given in our next definition.

**Definition 19.** We say that  $P$  is well-defined if for every finite-state system  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  and every  $f \in F$  the following conditions are satisfied:

- There are effectively definable formulae  $\Xi_f, \Gamma_f$  of the logic  $\mathcal{L}(\langle\langle\alpha\rangle\rangle, \mathbf{EF})$  such that for every process  $g$  where  $\mathcal{A}(g) \subseteq \mathcal{A}$  we have that  $g \models \Xi_f$  iff  $(f, g) \in P$ , and  $g \models \Gamma_f$  iff  $(g, f) \in P$ .
- There is a polynomial  $\mathcal{P}$  (in two variables) such that for every finite-state system  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  the set  $\{\Xi_f, \Gamma_f \mid f \in F\}$  can be computed, and the relation  $P \cap (F \times F)$  can be decided, in time  $\mathcal{O}(2^{\mathcal{P}(|F|, |\mathcal{A}|)})$ .

Note that the  $T$ ,  $D$ ,  $F$ , and  $R$  preorders are clearly well-defined. However, the  $S$  preorder is (provably) not well-defined. Nevertheless, our results *do* apply to possible-futures equivalence, as we shall see in Remark 24.

**Lemma 20.** If  $P$  is well-defined, then the relation  $\sqsubseteq_i$  over  $F \times 2^F$  can be computed in time which is exponential in  $n$  and polynomial in  $i$ .

#### 4.1 Encoding *PQ* Preorder into Modal Logic

**Definition 21.** For all  $i \in \mathbb{N}_0$ ,  $f \in F$ , and  $M \subseteq F$  we define the sets

- $\mathcal{F}(f, \sqsubseteq_i) = \{M \subseteq F \mid f \sqsubseteq_i M\}$
- $\mathcal{F}(\sqsubseteq_i, M) = \{f \in F \mid f \sqsubseteq_i M\}$ .

For all  $f \in F$  and  $k \in \mathbb{N}_0$  we define the formulae  $\Phi_{f,k}$ ,  $\Psi_{f,k}$ , and  $\Theta_{f,k}$  inductively as follows:

- $\Phi_{f,0} = \Xi_f, \Psi_{f,0} = \Gamma_f$
- $\Theta_{f,k} = \Phi_{f,k} \wedge \Psi_{f,k}$

- $\Phi_{f,k+1} = \Xi_f \wedge (\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}) \wedge (\bigwedge_{f \stackrel{\circ}{\rightarrow} f'} (\bigvee_{M \in \mathcal{F}(f', \sqsubseteq_k)} (\bigwedge_{f'' \in M} \langle\langle \alpha \rangle\rangle \Theta_{f'',k})))$
- $\Psi_{f,k+1} = \Gamma_f \wedge (\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}) \wedge \bigwedge_{\alpha \in \mathcal{A}_\tau} \llbracket \alpha \rrbracket (\bigvee_{f \stackrel{\circ}{\rightarrow} M} \bigvee_{f' \in \mathcal{F}(\sqsubseteq_k, M)} \Theta_{f',k})$

The empty conjunction is equivalent to  $\mathbf{tt}$ , and the empty disjunction to  $\mathbf{ff}$ .

The  $\mathcal{F}(\dots)$  sets are effectively constructible in time exponential in  $n$  and polynomial in  $i$  (Lemma 20), hence the  $\Phi_{f,k}, \dots$ , formulae are effectively constructible too.

**Theorem 22.** *Let  $g$  be an (arbitrary) process such that  $\mathcal{A}(g) \subseteq \mathcal{A}$ . Then for all  $f \in F$  and  $k \in \mathbb{N}_0$  we have the following:*

- (a)  $g \models \Phi_{f,0}$  iff  $f \sqsubseteq_0 g$ ; further,  $g \models \Phi_{f,k+1}$  iff  $f \sqsubseteq_{k+1} g$  and for each  $g \rightarrow^* g'$  there is  $f' \in F$  such that  $g' \sim_k f'$ .
- (b)  $g \models \Psi_{f,0}$  iff  $g \sqsubseteq_0 f$ ; further,  $g \models \Psi_{f,k+1}$  iff  $g \sqsubseteq_{k+1} f$  and for each  $g \rightarrow^* g'$  there is  $f' \in F$  such that  $g' \sim_k f'$ .
- (c)  $g \models \Theta_{f,0}$  iff  $g \stackrel{\circ}{=} f$ ; further,  $g \models \Theta_{f,k+1}$  iff  $f \sim_{k+1} g$  and for each  $g \rightarrow^* g'$  there is  $f' \in F$  such that  $g' \sim_k f'$ .

**Corollary 23.** *Let  $g$  be an (arbitrary) process such that  $\mathcal{A}(g) \subseteq \mathcal{A}$ , and let  $f \in F$ . Then the following two conditions are equivalent:*

- (a)  $g \sim f$  and for every  $g \rightarrow^* g'$  there is some  $f' \in F$  such that  $g' \sim f'$ .
- (b)  $g \models \Theta_{f,n2^n} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n2^n})$ .

Note that the size of the circuit representing the formula  $\Theta_{f,n2^n} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n2^n})$  is exponential in  $n$  and can be constructed in exponential time.

*Remark 24.* As we already mentioned, the  $S$  preorder is not well-defined, because trace equivalence with a given finite-state process  $f$  is not expressible in modal logic (even monadic second order logic is (provably) not sufficiently powerful to express that a process can perform every trace over a given finite alphabet). Nevertheless, in our context it suffices to express the condition of *full trace equivalence* with  $f$ , which is achievable. So, full possible-futures equivalence with  $f$  is expressed by the formula  $\Theta_{f,n2^n} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n2^n})$  where for every  $f' \in F$  we define  $\Xi_{f'}$  and  $\Gamma_{f'}$  to be the formula which expresses full trace equivalence with  $f'$ . This “trick” can be used also for other trace-like equivalences where the associated  $P$  is not well-defined.

## 5 Model checking lossy channel systems

In this section we show that the model checking of  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha)$  formulae is decidable for lossy channel systems (LCS’s). This result was inspired by [6] and can be seen as a natural extension of known results.

We refer to [1, 29] for motivations and definitions on LCS’s. Here we only need to know that a *configuration*  $\sigma$  of a LCS  $C$  is a pair  $\langle q, w \rangle$  of a control state  $q$  from some finite set  $Q$  and a finite word  $w \in \Sigma^*$  describing the current contents of the channel (for simplicity we assume a single channel). Here  $\Sigma = \{a, b, \dots\}$  is a finite alphabet of messages. The behavior of  $C$  is given by a transition system  $\mathcal{T}_C$  where steps  $\sigma \rightarrow \sigma'$

describe how the configuration can evolve. In the rest of this section, we assume a fixed LCS  $C$ .

Saying that the system is *lossy* means that messages can be lost while they are in the channel. This is formally captured by introducing an ordering between configurations: we write  $\langle q_1, w_1 \rangle \leq \langle q_2, w_2 \rangle$  when  $q_1 = q_2$  and  $w_1$  is a subword of  $w_2$  (i.e. one can obtain  $w_1$  by erasing some letters in  $w_2$ , possibly all letters, possibly none). Higman's lemma states that  $\leq$  is a well-quasi-ordering (a *wqo*), i.e. it is well-founded and any set of incomparable configurations is finite.

Losing messages in a configuration  $\sigma$  yields some  $\sigma'$  with  $\sigma' \leq \sigma$ . The crucial fact we shall use is that steps of LCS's are closed under losses:

**Lemma 25 (see [1, 29]).** *If  $\sigma \rightarrow \sigma'$  is a step of  $\mathcal{T}_C$ , then for all configurations  $\theta \geq \sigma$  and  $\theta' \leq \sigma'$ ,  $\theta \rightarrow \theta'$  is a step of  $\mathcal{T}_C$  too.*

We are interested in sets of configurations denoted by some simple expressions. For a configuration  $\sigma$  we let  $\uparrow\sigma$  denote the upward-closure of  $\sigma$ , i.e. the set  $\{\theta \mid \sigma \leq \theta\}$ . A *restricted set* is denoted by an expression  $\varrho$  of the form  $\uparrow\sigma - \uparrow\theta_1 - \dots - \uparrow\theta_n$  (for some configurations  $\theta_1, \dots, \theta_n$ ). This denotes an upward-closure minus some restrictions (the  $\uparrow\theta_i$ 's).

An expression  $\varrho$  is *trivial* if it denotes the empty set. Clearly  $\uparrow\sigma - \uparrow\theta_1 - \dots - \uparrow\theta_n$  is trivial iff  $\theta_i \leq \sigma$  for some  $i$ . A *constrained set* is a finite union of restricted sets, denoted by an expression  $\gamma$  of the form  $\varrho_1 \vee \dots \vee \varrho_m$ . Such an expression is *reduced* if no  $\varrho_i$  is trivial. For a set  $S$  of configurations,  $Pre(S) = \{\sigma \mid \exists \theta \in S, \sigma \rightarrow \theta\}$  is the set of (immediate) predecessors of configurations in  $S$ .

**Lemma 26.** *Constrained sets are closed under intersection, complementation, and Pre. Furthermore, from reduced expressions  $\gamma$ ,  $\gamma_1$  and  $\gamma_2$ , one can compute reduced expressions for  $\gamma_1 \wedge \gamma_2$ ,  $\neg\gamma$ , and  $Pre(\gamma)$ .*

We can now compute the set of configurations that satisfy an **EU** formula:

**Lemma 27.** *Let  $S_1$  and  $S_2$  be two constrained sets. Then the set  $S$  of configurations that satisfy  $S_1$  **EU**  $S_2$  is constrained too. Furthermore, from reduced expressions for  $S_1$  and  $S_2$ , one can compute a reduced expression for  $S$ .*

By combining Lemma 26 and Lemma 27, we obtain the result we were aiming at:

**Corollary 28.** *Let  $\varphi$  be a modal formula in  $\mathcal{L}(\mathbf{EX}, \mathbf{EU})$ . The set of configurations that satisfy  $\varphi$  is a constrained set, and one can compute a reduced expression for this set.*

**Theorem 29.** *The model checking problem for  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha)$  formulae is decidable for lossy channel systems.*

## 6 Applications

*A Note on Semantic Quotients.* Let  $\mathcal{T} = (S, \rightarrow, \mathcal{A})$  be a transition system,  $g \in S$ , and  $\sim$  a process equivalence. Let  $Reach(g) = \{s \in S \mid g \rightarrow^* s\}$ . The  $\sim$ -quotient of  $g$  is

the process  $[g]$  of the transition system  $(Reach(g)/\sim, \rightarrow, \mathcal{A})$  where  $[s] \xrightarrow{\alpha} [t]$  iff there are  $s', t' \in Reach(g)$  such that  $s \sim s', t \sim t'$ , and  $s' \xrightarrow{\alpha} t'$ .

For most (if not all) of the existing process equivalences we have that  $s \sim [s]$  for every process  $s$  (see [17, 18]). In general, the class of temporal properties preserved under  $\sim$ -quotients is larger than the class of  $\sim$ -invariant properties [18]. Hence,  $\sim$ -quotients are rather robust descriptions of the original systems. Some questions related to formal verification can be answered by examining the properties of  $\sim$ -quotients, which is particularly advantageous if the  $\sim$ -quotient is finite (so far, mainly the bisimilarity-quotients have been used for this purpose). This raises two natural problems:

- (a) Given a process  $g$  and an equivalence  $\sim$ , is the  $\sim$ -quotient of  $g$  finite?
- (b) Given a process  $g$ , an equivalence  $\sim$ , and a finite-state process  $f$ , is  $f$  the  $\sim$ -quotient of  $g$ ?

The question (a) is known as *the strong regularity problem* (see, e.g., [16] where it is shown that strong regularity wrt. simulation equivalence is decidable for one-counter nets). For bisimulation-like equivalences, the question (a) coincides with the standard regularity problem.

Using the results of previous sections, the problem (b) is reducible to the model-checking problem with the logic  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ . Let  $\mathcal{F} = (F, \rightarrow, \mathcal{A})$  be a finite state system and  $\sim$  an MTB or PQ equivalence. Further, let us assume that the states of  $\mathcal{F}$  are pairwise non-equivalent (this can be effectively checked). Consider the formula

$$\varrho_f \equiv \xi_f \wedge \bigwedge_{f' \in F} \mathbf{EF} \xi_{f'} \wedge \bigwedge_{\substack{f' \xrightarrow{\alpha} f'' \\ (\text{in } \mathcal{F})}} \mathbf{EF} (\xi_{f'} \wedge \mathbf{EX}_\alpha \xi_{f''}) \wedge \bigwedge_{\substack{f' \not\xrightarrow{\alpha} f'' \\ (\text{in } \mathcal{F})}} \mathbf{AG} (\xi_{f'} \Rightarrow \mathbf{AX}_\alpha \neg \xi_{f''})$$

where  $\xi_f$  is the formula expressing full  $\sim$ -equivalence with  $f$ . It is easy to see that for every process  $g$  s.t.  $\mathcal{A}(g) \subseteq \mathcal{A}(f)$  we have that  $g \models \varrho_f$  iff  $f$  is the  $\sim$ -quotient of  $g$ .

Observe that if the problem (b) above is decidable for a given class of processes, then the problem (a) is semidecidable for this class. So, for all those models where model-checking with the logic  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  is decidable we have that the positive subcase of the strong regularity problem is semidecidable due to rather generic reasons, while establishing the semidecidability of the negative subcase is a model-specific part of the problem.

*Results for concrete process classes.* All of the so far presented results are applicable to those process classes where model-checking the relevant fragment of modal logic is decidable. In particular, model-checking  $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$  is decidable for

- pushdown processes. In fact, this problem is **PSPACE**-complete [36]. Moreover, the complexity of the model-checking algorithm depends on the size of the circuit which represents a given formula (rather than on the size of the formula itself) [37];
- PA (and in fact also PAD) processes [24, 22]. The best known complexity upper bound for this problem is non-elementary.
- lossy channel systems (see Section 5). Here the model-checking problem is of non-primitive recursive complexity.

From this we immediately obtain that the problem of full *MTB*-equivalence, where  $B$  is well-defined, is

- decidable in polynomial space for pushdown processes. For many concrete *MTB*-equivalences, this bound is optimal (for example, all bisimulation-like equivalences between pushdown processes and finite-state processes are **PSPACE**-hard [23]);
- decidable for PA and PAD processes;
- decidable for lossy channel systems. For most concrete *MTB*-equivalences, the problem is of nonprimitive recursive complexity (this can be easily derived using the results of [29]).

Similar results hold for *PQ*-equivalences where  $P$  is well-defined (for pushdown processes we obtain **EXSPACE** upper complexity bound). Finally, the remarks about the problems (a),(b) of the previous paragraph also apply to the mentioned process classes.

## References

- [1] P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *I&C*, 127(2):91–101, 1996.
- [2] P.A. Abdulla, K. Čerāns, B. Jonsson, and Yih-Kuen Tsay. Algorithmic analysis of programs with well quasi-ordered domains. *I&C*, 160(1–2):109–127, 2000.
- [3] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *JACM*, 40(3):653–682, 1993.
- [4] J.C.M. Baeten and R.J. van Glabbeek. Another look at abstraction in process algebra. In *Proceedings of ICALP'87*, volume 267 of *LNCS*, pages 84–94. Springer, 1987.
- [5] A. Bouajjani. Languages, rewriting systems, and verification of infinite-state systems. In *Proceedings of ICALP'2001*, volume 2076 of *LNCS*, pages 24–39. Springer, 2001.
- [6] A. Bouajjani and R. Mayr. Model-checking lossy vector addition systems. In *Proceedings of STACS'99*, volume 1563 of *LNCS*, pages 323–333. Springer, 1999.
- [7] M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *TCS*, 59(1–2):115–131, 1988.
- [8] O. Burkart, 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*, pages 545–623. Elsevier, 2001.
- [9] J. Esparza and M. Nielsen. Decidability issues for Petri nets — a survey. *Journal of Information Processing and Cybernetics*, 30(3):143–160, 1994.
- [10] A. Finkel and Ph. Schnoebelen. Well structured transition systems everywhere! *TCS*, 256(1–2):63–92, 2001.
- [11] Y. Hirshfeld and M. Jerrum. Bisimulation equivalence is decidable for normed process algebra. In *Proceedings of ICALP'99*, volume 1644 of *LNCS*, pages 412–421. Springer, 1999.
- [12] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *TCS*, 158(1–2):143–159, 1996.
- [13] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimulation equivalence of normed basic parallel processes. *MSCS*, 6(3):251–259, 1996.
- [14] P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *TCS*, 148(2):281–301, 1995.
- [15] P. Jančar, A. Kučera, and R. Mayr. Deciding bisimulation-like equivalences with finite-state processes. *TCS*, 258(1–2):409–433, 2001.

- [16] P. Jančar, A. Kučera, and F. Moller. Simulation and bisimulation over one-counter processes. In *Proceedings of STACS'2000*, volume 1770 of *LNCS*, pages 334–345. Springer, 2000.
- [17] A. Kučera. On finite representations of infinite-state behaviours. *IPL*, 70(1):23–30, 1999.
- [18] A. Kučera and J. Esparza. A logical viewpoint on process-algebraic quotients. *JLC*, 13(6):863–880, 2003.
- [19] A. Kučera and P. Jančar. Equivalence-checking with infinite-state systems: Techniques and results. In *Proceedings of SOFSEM'2002*, volume 2540 of *LNCS*, pages 41–73. Springer, 2002.
- [20] A. Kučera and R. Mayr. A generic framework for checking semantic equivalences between pushdown automata and finite-state automata. In *Proceedings of IFIP TCS'2004*. Kluwer, 2004. To appear.
- [21] A. Kučera and Ph. Schnoebelen. A general approach to comparing infinite-state systems with their finite-state specifications. Technical report FIMU-RS-2004-05, Faculty of Informatics, Masaryk University, 2004.
- [22] D. Lugiez and Ph. Schnoebelen. The regular viewpoint on PA-processes. *TCS*, 274(1–2):89–115, 2002.
- [23] R. Mayr. On the complexity of bisimulation problems for pushdown automata. In *Proceedings of IFIP TCS'2000*, volume 1872 of *LNCS*, pages 474–488. Springer, 2000.
- [24] R. Mayr. Decidability of model checking with the temporal logic EF. *TCS*, 256(1–2):31–62, 2001.
- [25] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- [26] M. Müller-Olm. Derivation of characteristic formulae. *ENTCS*, 18, 1998.
- [27] J. Parrow and P. Sjödin. Multiway synchronization verified with coupled simulation. In *Proceedings of CONCUR'92*, volume 630 of *LNCS*, pages 518–533. Springer, 1992.
- [28] J. Parrow and P. Sjödin. The complete axiomatization of cs-congruence. In *Proceedings of STACS'94*, volume 775 of *LNCS*, pages 557–568. Springer, 1994.
- [29] Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *IPL*, 83(5):251–261, 2002.
- [30] G. Sénizergues.  $L(A)=L(B)$ ? Decidability results from complete formal systems. *TCS*, 251(1–2):1–166, 2001.
- [31] J. Srba. Roadmap of infinite results. *EATCS Bulletin*, 78:163–175, 2002.
- [32] B. Steffen and A. Ingólfssdóttir. Characteristic formulae for processes with divergence. *I&C*, 110(1):149–163, 1994.
- [33] R.J. van Glabbeek. The linear time—branching time spectrum II: The semantics of sequential systems with silent moves. In *Proceedings of CONCUR'93*, volume 715 of *LNCS*, pages 66–81. Springer, 1993.
- [34] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *JACM*, 43(3):555–600, 1996.
- [35] M. Voorhoeve and S. Mauw. Impossible futures and determinism. *IPL*, 80(1):51–58, 2001.
- [36] I. Walukiewicz. Model checking CTL properties of pushdown systems. In *Proceedings of FST&TCS'2000*, volume 1974 of *LNCS*, pages 127–138. Springer, 2000.
- [37] I. Walukiewicz. Private communication, September 2003.