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

Antonín Kučera [1]

Faculty of Informatics, Masaryk University,
Botanická 68a, CZ-60200 Brno,
Czech Republic

Philippe Schnoebelen [2]

LSV, ENS de Cachan & CNRS UMR 8643,
61, av. Pdt. Wilson, 94235 Cachan Cedex,
France.

Abstract

We introduce a generic family of behavioral relations for which the regular equivalence problem (i.e., comparing an arbitrary transition system to some finite-state specification) can be reduced to the 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.

Key words: verification, semantic equivalences, infinite-state systems, modal logic

---

Email addresses: `tony@fi.muni.cz` (Antonín Kučera),
`phs@lsv.ens-cachan.fr` (Philippe Schnoebelen).
URLs: `http://www.fi.muni.cz/usr/kucera` (Antonín Kučera),
`http://www.lsv.ens-cachan.fr/~phs/` (Philippe Schnoebelen).

[1] The first author is supported by the research centre "Institute for Theoretical Computer Science (ITI)", project No. 1M0021620808.
[2] The second author is supported by the ACI Sécurité & Informatique (project Persée) funded by the French Ministry of Research.

1 Introduction

Verification of infinite-state systems is a very active research field (see, e.g., [11,9,5,21,34] 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 modal 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 model checking problems. Regarding equivalence problems, positive decidability results exist mainly for strong bisimilarity (some milestones in the study include [3,15,14,16,13,33]). For other behavioral equivalences, results are usually negative.

1.1 Regular Equivalence Problem

Recently, the problem of comparing an infinite-state process g with its *finite-state* specification f has been identified as an important subcase [3] of the general equivalence checking problem [21]. 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, subprocess creation, or unbounded buffers, that make the model infinite-state. On the other 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 [21].

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 [12].

The second approach is even more general: one expresses equivalence with f via a formula φ_f of some modal logic $\mathcal{L}$. φ_f is called a *characteristic formula* for f wrt. the given equivalence. This reduces regular equivalence problems

[3] 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.

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 μ-calculus [35,29]. Browne *et al.* constructed characteristic formulae wrt. bisimilarity and branching-bisimilarity in the logic CTL [8]. Unfortunately, CTL (or μ-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 [17]. This logic is substantially simpler than CTL, and its associated model-checking problem is decidable in many classes of infinite-state systems (including PA, lossy channel systems, and pushdown automata) [27].

1.2 Our contribution

In Section 2 we introduce the notion of *full regular equivalence*. Compared to the "ordinary" regular equivalence discussed in previous paragraphs, 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). Then, we present a generic reduction of the full regular equivalence problem to the model checking problem for (essentially) the EF fragment of modal logic [4].

We offer two main reductions. The first reduction, presented in Section 3, applies to a family of equivalences defined via a "transfer property" (which means that the equivalence or preorder between a given pair of states can be transfered to their successors). This family includes bisimulation-like, simulation-like, and contrasimulation-like equivalences, which are abstracted and unified into a single notion of "MTB equivalence". The M, T, and B are parameters which hide the difference among the individual equivalences. Our reduction is generic in the sense that it works for an arbitrary MTB equivalence. The constructed modal formula is of exponential size, but can be efficiently represented by a circuit of polynomial size. This influences some of our complexity estimations presented later in Section 6.

The other reduction, presented in Section 4, applies to a family of equivalences based on sets of "enriched traces". A trace is a finite sequence of actions performable from a given state. Such a trace can be "enriched" by additional information about properties of states that are passed through along

[4] 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.

the trace. Similarly as in Section 3, we consider a single unified notion of "PS equivalence", where the P and S are parameters of the definition. The reduction is technically different from the one of Section 3, but there are some similarities. The constructed modal formula is of exponential size, even if it is represented by a circuit. Hence, the complexity results for PS equivalences are generally worse than the ones for MTB equivalences, which is consistent with the known complexity bounds for concrete models and equivalences (see Section 6 for more details).

The MTB and PS equivalences together cover virtually all process equivalences used in verification [36]. 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 models. Thus, as an important outcome we obtain that full regular equivalence is "more decidable and tractable" than regular equivalence. For example, regular trace equivalence is undecidable for BPA processes (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. At the same time, we should note that our generic algorithms based of reduction to EF model checking are not necessarily optimal for a given model. For example, it has been shown in [24] 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. See Section 2 and Section 6.2 for further comments.

A closer look at the presented reductions reveals that the constructions actually 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). Thus, we also obtain a refinement of the result presented in [8] which states that a characteristic formula wrt. branching bisimilarity is constructible in CTL.

Another contribution of this paper is a model-checking algorithm for the logic $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}, \mathbf{EU}_\alpha)$ and lossy channel systems given in Section 5. 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 EF model checking is already known).

Specific corollaries to our abstract results are summarized in Section 6.

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_τ.

Definition 1 *A* transition system *is a triple $\mathcal{T} = (S, \rightarrow, \mathcal{A})$ where S is a set of* states*, $\mathcal{A} \subset Act_\tau$ 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 $\stackrel{\alpha}{\Rightarrow} \subseteq S \times S$ as follows:

- $s \stackrel{\tau}{\Rightarrow} t$ iff there is a sequence of the form $s = p_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} p_k = t$ where $k \geq 0$;
- $s \stackrel{a}{\Rightarrow} t$ where $a \neq \tau$ iff there are p, q such that $s \stackrel{\tau}{\Rightarrow} p \xrightarrow{a} q \stackrel{\tau}{\Rightarrow} t$.

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 τ 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 *τ-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 2 (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 $\mathcal{F}$ such that $g' \sim f'$ (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 four states f, f', f'', f''' of Fig. 1 (right). Suppose that all transitions of a given infinite-state system g are labeled by a. Then regular trace equivalence to f means

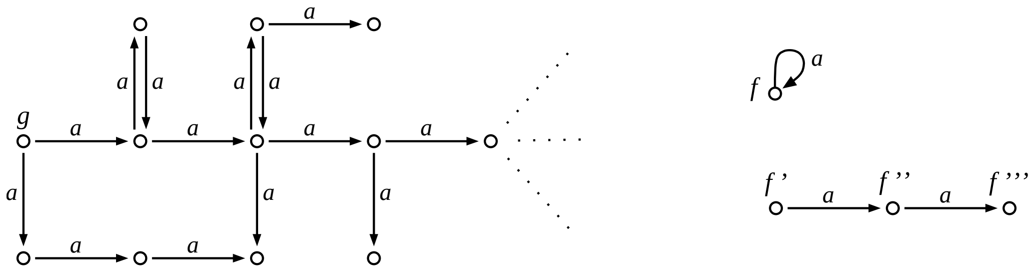

Fig. 1. Processes g and f.

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 two transitions (for example, the process g of Fig. 1 (left) is fully trace equivalent to f). 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 two parameterized families of process preorders and process equivalences. The first family of "MTB preorders/equivalences" corresponds to preorders/equivalences defined via a transfer property, such as bisimulation equivalence or simulation preorder/equivalence.

A *transfer* is one of the three operators on binary relations defined as follows (R denotes a binary relation):

- $\mathit{sim}(R) = R$,
- $\mathit{bisim}(R) = R \cap R^{-1}$,
- $\mathit{contrasim}(R) = R^{-1}$.

A *mode* is a subset of $\{\eta, d\}$ (the η and d are just two different symbols). A *basis* is an equivalence over processes satisfying the following property: whenever $(s, u) \in B$ and $s \stackrel{\tau}{\Rightarrow} t \stackrel{\tau}{\Rightarrow} u$, then also $(s, t) \in B$.

Definition 3 *Let $\mathcal{S}$ be a binary relation over processes and M a mode. A move $s \stackrel{\alpha}{\Rightarrow} t$ is* tightly $\mathcal{S}$-consistent *with M if either $\alpha = \tau$ and $s = t$, or there is a sequence $s = s_0 \stackrel{\tau}{\rightarrow} \cdots \stackrel{\tau}{\rightarrow} s_k \stackrel{\alpha}{\rightarrow} t_0 \stackrel{\tau}{\rightarrow} \cdots \stackrel{\tau}{\rightarrow} t_\ell = t$, 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 $0 \le i, j \le k$;
(2) if $d \in M$, then $(t_i, t_j) \in \mathcal{S}$ for all $0 \le i, j \le \ell$.

The loose $\mathcal{S}$-consistency *of $s \stackrel{\alpha}{\Rightarrow} t$ 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 4 *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 $p \stackrel{\alpha}{\Rightarrow} p'$ there is a tightly (or loosely, resp.) $\mathcal{R}$-consistent move $q \stackrel{\alpha}{\Rightarrow} q'$ such that $(p', q') \in T(\mathcal{R})$.*

*We write $s \sqsubseteq t$ (or $s \preccurlyeq t$, 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 $s \preccurlyeq t$ and $t \preccurlyeq s$, resp.).*

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

Lemma 5 *We have that $\sqsubseteq\, =\, \preccurlyeq$ (and hence also $\sim\, =\, \approx$).*

PROOF. ($\sqsubseteq\, \subseteq\, \preccurlyeq$). We show that $\sqsubseteq$ is a loose MTB-relation. So, let $s \sqsubseteq t$ and let $s \stackrel{\alpha}{\Rightarrow} s'$ be a loosely $\sqsubseteq$-consistent move. If this move is also tightly $\sqsubseteq$-consistent, there must be (due to $s \sqsubseteq t$) a tightly (and hence also loosely) $\sqsubseteq$-consistent move $t \stackrel{\alpha}{\Rightarrow} t'$ where $(s', t') \in T(\sqsubseteq)$ and we are done immediately. If the move $s \stackrel{\alpha}{\Rightarrow} s'$ is only loosely $\sqsubseteq$-consistent, it is of the form $s = p_0 \stackrel{\tau}{\Rightarrow} p_k \stackrel{\alpha}{\rightarrow} q_0 \stackrel{\tau}{\Rightarrow} q_\ell = s'$, where $k, \ell \ge 0$, and

- if $\eta \in M$, then $s \sim p_k$;
- if $d \in M$, then $s' \sim q_0$.

Now consider the subsequence $x \stackrel{\alpha}{\Rightarrow} y$ of the sequence $s = p_0 \stackrel{\tau}{\Rightarrow} p_k \stackrel{\alpha}{\rightarrow} q_0 \stackrel{\tau}{\Rightarrow} q_\ell = s'$ where

- if $\eta \in M$, then $x = p_k$, otherwise $x = p_0 = s$;
- if $d \in M$, then $y = q_0$, otherwise $y = q_\ell = s'$.

Observe that $x \sim s$, $y \sim s'$, and the move $x \stackrel{\alpha}{\Rightarrow} y$ is *tightly* $\sqsubseteq$-consistent. Since

$x \sim s$ and $s \sqsubseteq t$, there is a tightly (and hence also loosely) $\sqsubseteq$-consistent move $t \stackrel{\alpha}{\Rightarrow} t'$ such that $(y, t') \in T(\sqsubseteq)$. Since $s' \sim y$, we have $(s', t') \in T(\sqsubseteq)$ as needed.

($\preccurlyeq \subseteq \sqsubseteq$). We show that $\preccurlyeq$ is a tight MTB-relation. Let $s \preccurlyeq t$ and let $s \stackrel{\alpha}{\Rightarrow} s'$ be a tightly $\preccurlyeq$-consistent move. Since $s \preccurlyeq t$, there is a loosely $\preccurlyeq$-consistent move $t \stackrel{\alpha}{\Rightarrow} t'$ such that $s' \preccurlyeq t'$. We prove that $t \stackrel{\alpha}{\Rightarrow} t'$ is in fact tightly $\preccurlyeq$-consistent. To do that, consider the relation $\mathcal{R}$ defined as follows: $(p, q) \in \mathcal{R}$ iff there are processes p_1, p_2, q_1, q_2 such that $p_1 \approx p_2 \approx q_1 \approx q_2$, $p_1 \stackrel{\tau}{\Rightarrow} p \stackrel{\tau}{\Rightarrow} p_2$, and $q_1 \stackrel{\tau}{\Rightarrow} q \stackrel{\tau}{\Rightarrow} q_2$. Observe that $\mathcal{R}$ is reflexive and symmetric. Further, $\preccurlyeq \subseteq \mathcal{R}$ which means that if we manage to prove that $\mathcal{R}$ is a loose MTB-relation, we can conclude that $\preccurlyeq = \mathcal{R}$. This suffices for our purposes, because then we can readily justify the tight $\preccurlyeq$-consistency of the move $t \stackrel{\alpha}{\Rightarrow} t'$ — all of the intermediate states we wish to be related by $\preccurlyeq$ are clearly related by $\mathcal{R}$. First, let us realize that $\mathcal{R} \subseteq B$ (here we need the defining property of B). Now let $(p, q) \in \mathcal{R}$ and let p_1, p_2, q_1, q_2 be the four processes which witness the membership of (p, q) to $\mathcal{R}$. Further, let $p \stackrel{\alpha}{\Rightarrow} p'$ be a loosely $\mathcal{R}$-consistent move. We need to show that there is an $\mathcal{R}$-consistent move $q \stackrel{\alpha}{\Rightarrow} q'$ such that $(p', q') \in T(\mathcal{R})$. Observe that the move $p_1 \stackrel{\tau}{\Rightarrow} p \stackrel{\alpha}{\Rightarrow} p'$ is also loosely $\mathcal{R}$-consistent, because $p_1 \stackrel{\tau}{\Rightarrow} p$ passes through states which are all mutually related by $\mathcal{R}$. As $p_1 \approx q_2$, there is a loosely $\preccurlyeq$-consistent (and hence also $\mathcal{R}$-consistent) move $q_2 \stackrel{\alpha}{\Rightarrow} q'$ such that $(p', q') \in T(\preccurlyeq)$ (hence also $(p', q') \in T(\mathcal{R})$). Since $q \stackrel{\tau}{\Rightarrow} q_2$ passes through states which are mutually related by $\mathcal{R}$, the move $q \stackrel{\tau}{\Rightarrow} q_2 \stackrel{\alpha}{\Rightarrow} q'$ is also loosely $\mathcal{R}$-consistent and we are done. $\square$

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 T. Additional conditions which must be satisfied by equivalent processes can be specified by an appropriately defined basis. For example, we can put B to be *true*, *ready*, *terminate*, or *simulate*, where

- $(s, t) \in$ *true* for all s and t;
- $(s, t) \in$ *ready* iff $\{a \in Act_\tau \mid \exists s' : s \stackrel{a}{\Rightarrow} s'\} = \{a \in Act_\tau \mid \exists t' : t \stackrel{a}{\Rightarrow} t'\}$;
- $(s, t) \in$ *terminate* iff s and t are either both terminating, or both non-terminating (a process p is terminating iff $p \stackrel{\alpha}{\Rightarrow} p'$ implies $\alpha = \tau$ and p cannot perform an infinite sequence of τ-transitions).
- $(s, t) \in$ *simulate* iff s and t are *simulation equivalent* (see below).

The mode specifies the level of "control" over the states that are passed through by $\stackrel{\alpha}{\Rightarrow}$ transitions. In particular, by putting $T =$ *bisim*, $B =$ *true*, and choosing M to be $\emptyset$, $\{\eta\}$, $\{d\}$, or $\{\eta, d\}$, one obtains weak bisimilarity [28], η-bisimilarity [4], delay-bisimilarity, and branching bisimilarity [37], respectively.[5] "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 $T = sim$, $B = true$, and $M = \emptyset$, we obtain weak simulation equivalence; and by redefining B to *ready* and *simulate* we yield ready simulation equivalence and 2-nested simulation equivalence, respectively. The equivalence where $T = contrasim$, $B = true$, and $M = \emptyset$ is known as contrasimulation (see, e.g., [38]).

Remark 6 *Contrasimulation can also be seen as a generalization of coupled simulation [30,31], which was defined only for the subclass of divergence-free processes (where it coincides with contrasimulation). It is worth noting that contrasimulation coincides with strong bisimilarity on the subclass of τ-free processes (to see this, realize that one has to consider the moves $s \stackrel{\tau}{\Rightarrow} s$ even if s is τ-free). This is (intuitively) the reason why contrasimulation has some nice properties also in the presence of silent moves.*

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 7 *For every $k \in \mathbb{N}_0$, the binary relations $\sqsubseteq_k$, $\sim_k$, $\preccurlyeq_k$, 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 $s \stackrel{a}{\Rightarrow} s'$ there is some tightly $\sqsubseteq_k$-consistent move $t \stackrel{a}{\Rightarrow} t'$ such that $(s',t') \in T(\sqsubseteq_k)$.*

The $\preccurlyeq_k$ relations are defined in the same way, but we require only loose $\preccurlyeq_k$-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 $s \preccurlyeq_k t$ and $t \preccurlyeq_k s$.

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

Lemma 8 *Let $\mathcal{F} = (F, \rightarrow, \mathcal{A})$ be a finite-state system with n states. Then $\sqsubseteq_{n^2-1} = \sqsubseteq_{n^2} = \sqsubseteq = \preccurlyeq = \preccurlyeq_{n^2-1} = \preccurlyeq_{n^2}$, where all of the relations are considered as being restricted to $F \times F$.*

[5] Our definition of MTB equivalence does not directly match the definitions of η-, delay-, and branching bisimilarity that one finds in the literature. However, it is easy to show that one indeed yields exactly these equivalences.

PROOF. Since every binary relation over F has at most n^2 elements and $\sqsubseteq_{k+1}$ refines $\sqsubseteq_k$ for each k, we immediately obtain $\sqsubseteq_{n^2-1} = \sqsubseteq_{n^2}$. This means that $\sqsubseteq_{n^2}$ is a tight MTB-relation and hence $\sqsubseteq_{n^2} = \sqsubseteq$. For the same reason, $\preccurlyeq_{n^2-1} = \preccurlyeq_{n^2} = \preccurlyeq$. Note that $\sqsubseteq = \preccurlyeq$ by Lemma 5. $\square$

Theorem 9 *Let $\mathcal{F} = (F, \to, \mathcal{A})$ 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 $g \to^* g'$ there is some $f' \in F$ such that $g' \sim f'$.
(b) $g \sim_{n^2} f$ and for every $g \to^* g'$ there is some $f' \in F$ such that $g' \sim_{n^2} f'$.
(c) $g \approx_{n^2} f$ and for every $g \to^* g'$ there is some $f' \in F$ such that $g' \approx_{n^2} f'$.

PROOF. Clearly $(a) \Rightarrow (b)$ and $(a) \Rightarrow (c)$ (for the second implication we need Lemma 5). We prove that $(b) \Rightarrow (a)$ and $(c) \Rightarrow (a)$.

$(b) \Rightarrow (a)$: Let $G = \{g' \mid g \to^* g'\}$. We show that the relation $\sqsubseteq_{n^2}$ restricted to $(G \times F) \cup (F \times G)$ is a tight MTB-relation. So, let $\bar{g} \in G$, $\bar{f} \in F$ be processes such that

- (i) $\bar{g} \sqsubseteq_{n^2} \bar{f}$. Let $\bar{g} \overset{\alpha}{\Rightarrow} \bar{g}'$ be a tightly $\sqsubseteq_{n^2}$-consistent move. By definition of $\sqsubseteq_{n^2}$, there is a tightly $\sqsubseteq_{n^2-1}$-consistent move $\bar{f} \overset{\alpha}{\Rightarrow} \bar{f}'$ such that $(\bar{g}', \bar{f}') \in T(\sqsubseteq_{n^2-1})$. First, realize that the move $\bar{f} \overset{\alpha}{\Rightarrow} \bar{f}'$ is also tightly $\sqsubseteq_{n^2}$-consistent, because $\sqsubseteq_{n^2-1} = \sqsubseteq_{n^2}$ over $F \times F$ (see Lemma 8). Now we prove that $(\bar{g}', \bar{f}') \in T(\sqsubseteq_{n^2})$. Since $\bar{g}'$ is reachable from g, there is some $f' \in F$ such that $\bar{g}' \sim_{n^2} f'$. As $(\bar{g}', \bar{f}') \in T(\sqsubseteq_{n^2-1})$ and $\bar{g}' \sim_{n^2} f'$, we have that $(f', \bar{f}') \in T(\sqsubseteq_{n^2-1})$. However, this means that $(f', \bar{f}') \in T(\sqsubseteq_{n^2})$ by Lemma 8. As $(f', \bar{f}') \in T(\sqsubseteq_{n^2})$ and $\bar{g}' \sim_{n^2} f'$, we obtain $(\bar{g}', \bar{f}') \in T(\sqsubseteq_{n^2})$ as needed.
- (ii) $\bar{f} \sqsubseteq_{n^2} \bar{g}$. Let $\bar{f} \overset{\alpha}{\Rightarrow} \bar{f}'$ be a tightly $\sqsubseteq_{n^2}$-consistent move. Then there is (by definition of $\sqsubseteq_{n^2}$) a tightly $\sqsubseteq_{n^2-1}$-consistent move $\bar{g} \overset{\alpha}{\Rightarrow} \bar{g}'$ such that $(\bar{f}', \bar{g}') \in T(\sqsubseteq_{n^2-1})$. Now it suffices to show that
  - (1) the move $\bar{g} \overset{\alpha}{\Rightarrow} \bar{g}'$ is in fact tightly $\sqsubseteq_{n^2}$-consistent. This is justified by observing that for any two states g_1, g_2 which appear along the move $\bar{g} \overset{\alpha}{\Rightarrow} \bar{g}'$ we have that $g_1 \sim_{n^2-1} g_2$ implies $g_1 \sim_{n^2} g_2$. To see this, realize that g_1, g_2 are reachable from g and hence there are some $f_1, f_2 \in F$ such that $g_1 \sim_{n^2} f_1$ and $g_2 \sim_{n^2} f_2$. Since $f_1 \sim_{n^2} g_1 \sim_{n^2-1} g_2 \sim_{n^2} f_2$, we obtain $f_1 \sim_{n^2-1} f_2$ and hence also $f_1 \sim_{n^2} f_2$ by Lemma 8. Now $g_1 \sim_{n^2} f_1 \sim_{n^2} f_2 \sim_{n^2} g_2$, thus $g_1 \sim_{n^2} g_2$.
  - (2) $(\bar{f}', \bar{g}') \in T(\sqsubseteq_{n^2})$. This follows from $(\bar{f}', \bar{g}') \in T(\sqsubseteq_{n^2-1})$ by using the same argument as in (i).

$(c) \Rightarrow (a)$: Using the same technique as above, one can prove that $\preccurlyeq_{n^2}$ restricted to $(G \times F) \cup (F \times G)$ is a loose MTB-relation. The claim then follows by applying Lemma 5. $\square$

In this section we show that the conditions (b) and (c) of Theorem 9 can be expressed in modal logic. Let us consider a class of modal formulae defined by the following abstract syntax equation (where α ranges over Act_τ):

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

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

- $s \models \mathtt{tt}$ 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 $s \stackrel{\tau}{\Rightarrow} s'$ such that $s' \models \varphi$.
- $s \models \varphi_1\,\mathbf{EU}\,\varphi_2$ iff either $s \models \varphi_2$, or there is a sequence $s = s_0 \xrightarrow{a_1} \cdots \xrightarrow{a_m} s_m$, where $m \geq 0$, $a_i \in Act_\tau$ for every $1 \leq i \leq m$, $s_i \models \varphi_1$ for all $0 \leq i < m$, and $s_m \models \varphi_2$.
- $s \models \varphi_1\,\mathbf{EU}_\alpha\,\varphi_2$ iff either $\alpha = \tau$ and $s \models \varphi_2$, or there is a sequence $s = s_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} s_m \xrightarrow{\alpha} s'$, 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 **EF** is **AG**, defined by $\mathbf{AG}\,\varphi \equiv \neg\,\mathbf{EF}\,\neg\varphi$.

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

- $\mathcal{L}(\mathbf{EX}_\alpha)$ is the well-known Hennessy-Milner logic [28];
- $\mathcal{L}(\mathbf{EU}_\alpha)$ is the logic proposed by de Nicola and Vaandrager in [10] which modally characterizes branching bisimilarity;
- $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ is the logic used in [17] to construct characteristic formulae wrt. full and weak bisimilarity for finite-state systems. As opposed to other modal logics, the model-checking problem with $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ is decidable for many classes of infinite-state systems (e.g., BPA, BPP, and PA process algebras, pushdown automata, lossy channel systems, etc.)

Let $\sim$ be an *MTB* equivalence. Our aim is to show that for every finite f there are formulae φ_f of $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$ and ψ_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 9. Clearly such formulae cannot always exist without some additional assumptions about the basis B. Actually, all we need is to assume that the *full* B-equivalence 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 Ξ^t_f and Ξ^ℓ_f 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 is fully B-equivalent to f iff $g \models \Xi^t_f$ iff $g \models \Xi^\ell_f$. Note that g is fully B-equivalent to f iff $(g, f) \in B$ and for every $g \to^* g'$ there is $f' \in F$ such that $(g', f') \in B$. At first glance, it seems that full B-equivalence with f is harder to express that just B-equivalence with f. In fact, the opposite holds—if ϱ_f is the formula expressing the B-equivalence with a given $f \in F$, then $\varrho_f \wedge \mathbf{AG} \bigvee_{f' \in F} \varrho_{f'}$ is the formula expressing the *full* B-equivalence with the state f. On the other hand, there are B's for which full B-equivalence with a given $f \in F$ is expressible in $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$ and $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$, but B-equivalence with f is *not* expressible. A concrete example is the *simulate* basis introduced in the previous subsection. As we shall see, full simulation equivalence with a given f is expressible in $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$ and $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$, while simulation equivalence with f is not (otherwise we would immediately derive a contradiction with known decidability results [21]). Since we are also interested in complexity issues, we further assume that the formulae Ξ^t_f and Ξ^ℓ_f 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 10 *We say that a basis B is* well-defined *if there is a polynomial $\mathcal{P}$ (in two variables) such that for every finite-state system $\mathcal{F} = (F, \to, \mathcal{A})$ the set $\{\Xi^t_f, \Xi^\ell_f \mid f \in F\}$ can be computed, and the relation $B \cap (F \times F)$ can be decided, in time $\mathcal{O}(\mathcal{P}(|F|, |\mathcal{A}|))$.*

Remark 11 *In fact, the Ξ^t_f formulae are only required for the construction of φ_f, and the Ξ^ℓ_f formulae are required only for the construction of ψ_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 Ξ^t_f and Ξ^ℓ_f families, which means that for some MTB equivalences we can construct only one of the φ_f and ψ_f formulae. A concrete example is the terminate basis of the previous section, which is definable in $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ but not in $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$.*

For the rest of this section, we fix some MTB-equivalence $\sim$ where B is well-defined, and a finite-state system $\mathcal{F} = (F, \to, \mathcal{A})$ 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 $s = p_0 \xrightarrow{\tau} \cdots p_k \xrightarrow{\alpha} q_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} q_m$, 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 $s = p_0 \xrightarrow{\tau} \cdots p_k \xrightarrow{\alpha} q_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} q_m$, 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 12 *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 $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}_\tau)$, respectively:*

PROOF. It suffices to realize that

$$\langle \alpha, \varphi_\eta, \varphi_d \rangle^t \varphi \;\equiv\; \begin{cases} \varphi_\eta \wedge (\varphi_\eta \,\mathbf{EU}_\alpha (\varphi_d \,\mathbf{EU}_\tau (\varphi_d \wedge \varphi))) & \text{if } \alpha \neq \tau \\[1ex] (\varphi_\eta \wedge (\varphi_\eta \,\mathbf{EU}_\alpha (\varphi_d \,\mathbf{EU}_\tau (\varphi_d \wedge \varphi)))) \vee \varphi & \text{if } \alpha = \tau \end{cases}$$

$$\langle \alpha, \varphi_\eta, \varphi_d \rangle^\ell \varphi \;\equiv\; \begin{cases} \varphi_\eta \wedge \mathbf{EF}_\tau (\varphi_\eta \wedge \mathbf{EX}_\alpha (\varphi_d \wedge \mathbf{EF}_\tau (\varphi_d \wedge \varphi))) & \text{if } \alpha \neq \tau \\[1ex] (\varphi_\eta \wedge \mathbf{EF}_\tau (\varphi_\eta \wedge \mathbf{EX}_\alpha (\varphi_d \wedge \mathbf{EF}_\tau (\varphi_d \wedge \varphi)))) \vee \varphi & \text{if } \alpha = \tau \end{cases}$$

$\square$

Since the conditions (b) and (c) of Theorem 9 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 $\langle \alpha, \varphi_\eta, \varphi_d \rangle^\ell$, Ξ_f denotes either Ξ_f^t or Ξ_f^ℓ, $\circeq_k$ denotes either $\sim_k$ or $\approx_k$, and $\leqslant_k$ denotes either $\sqsubseteq_k$ or $\preccurlyeq_k$, respectively. Moreover, we write $s \xrightarrow{\alpha,k} t$ to denote that there is either a tightly $\sqsubseteq_k$-consistent move $s \stackrel{\alpha}{\Rightarrow} t$, or a loosely $\preccurlyeq_k$-consistent move $s \stackrel{\alpha}{\Rightarrow} t$, respectively.

Definition 13 *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:*

$$\begin{aligned}
\Phi_{f,0} &= \Psi_{f,0} = \Xi_f \\
\Theta_{f,k} &= \Phi_{f,k} \wedge \Psi_{f,k} \\
\Phi_{f,k+1} &= \Xi_f \;\wedge\; \Big(\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}\Big) \;\wedge\; \bigwedge_{f \xrightarrow{\alpha,k} f'} \;\bigvee_{f_1, f_2 \in F} \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle \xi_{f',k} \\
\Psi_{f,k+1} &= \Xi_f \;\wedge\; \Big(\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}\Big) \;\wedge\; \bigwedge_{\substack{\alpha \in \mathcal{A}_\tau \\ f_1, f_2 \in F}} \Big([\alpha, \varphi_{f_1,k}, \psi_{f_2,k}] \bigvee_{f \xrightarrow{\alpha,k} f'} \varrho_{f',k}\Big)
\end{aligned}$$

where

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

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

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 $f \leqslant_k g$, and $g \models \Psi_{f,k}$ iff $g \leqslant_k f$. However, this is (provably) *not achievable*—the $\leqslant_k$ 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 14 *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* $f \leqslant_0 g$*; further,* $g \models \Phi_{f,k+1}$ *iff* $f \leqslant_{k+1} g$ *and for each* $g \rightarrow^* g'$ *there is* $f' \in F$ *such that* $g' \mathrel{\mathring{=}}_k f'$*.*
(b) $g \models \Psi_{f,0}$ *iff* $g \leqslant_0 f$*; further,* $g \models \Psi_{f,k+1}$ *iff* $g \leqslant_{k+1} f$ *and for each* $g \rightarrow^* g'$ *there is* $f' \in F$ *such that* $g' \mathrel{\mathring{=}}_k f'$*.*
(c) $g \models \Theta_{f,0}$ *iff* $g \mathrel{\mathring{=}}_0 f$*; further,* $g \models \Theta_{f,k+1}$ *iff* $f \mathrel{\mathring{=}}_{k+1} g$ *and for each* $g \rightarrow^* g'$ *there is* $f' \in F$ *such that* $g' \mathrel{\mathring{=}}_k f'$*.*

PROOF. We prove (a) and (b) by induction on k (the (c) follows immediately then). The base case when $k = 0$ is trivial. It remains to show the inductive step of (a) and (b).

(a) We start with the "$\Leftarrow$" direction. Since $f \leqslant_{k+1} g$ and for each $g \rightarrow^* g'$ there is $f' \in F$ such that $g' \mathrel{\mathring{=}}_k f'$, we can apply the induction hypothesis to conclude that $g \models \Xi_f \wedge (\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k})$. It remains to prove that g satisfies also the formula

$$\bigwedge_{f \overset{\alpha,k}{\longrightarrow} f'} \Big(\bigvee_{f_1, f_2 \in F} \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle \xi_{f',k} \Big).$$

To see this, realize that for each $f \overset{\alpha,k}{\longrightarrow} f'$ there is some $g \overset{\alpha,k}{\longrightarrow} g'$ such that $(f', g') \in T(\leqslant_k)$. Since g, g' are reachable from g, there are some $f_1, f_2 \in F$ such that $g \mathrel{\mathring{=}}_k f_1$ and $g' \mathrel{\mathring{=}}_k f_2$. As $g \overset{\alpha,k}{\longrightarrow} g'$, we can apply the induction hypothesis and conclude that $g \models \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle \xi_{f',k}$. This works for arbitrary $f \overset{\alpha,k}{\longrightarrow} f'$, hence $g \models \bigwedge_{f \overset{\alpha,k}{\longrightarrow} f'} (\bigvee_{f_1,f_2 \in F} \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle \xi_{f',k})$ as needed.

For the "⇒" direction, let us suppose that $g \models \Xi_f \wedge (\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k})$. Since $g \models \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$, we can apply the induction hypothesis to conclude that for every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \doteqdot_k f'$. It remains to show that $f \leqslant_{k+1} g$. Clearly $(f, g) \in B$ because $g \models \Xi_f$. Let $f \stackrel{\alpha,k}{\rightarrow} f'$. As $g \models \bigvee_{f_1,f_2 \in F} \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle \xi_{f',k}$, there are $f_1, f_2 \in F$ such that $g \models \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle \xi_{f',k}$. By applying the induction hypothesis we obtain that there is $g \stackrel{\alpha,k}{\rightarrow} g'$ such that $g' \models \xi_{f',k}$, which means $(f', g') \in T(\leqslant_k)$.

(b) "⇐": Let us assume that $g \leqslant_{k+1} f$ and for each $g \rightarrow^* g'$ there is $f' \in F$ such that $g' \doteqdot_k f'$. Then $g \models \Xi_f \wedge (\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k})$ by induction hypothesis. Now let $\alpha \in \mathcal{A}_\tau$ and $f_1, f_2 \in F$. We show that $g \models [\alpha, \varphi_{f_1,k}, \psi_{f_2,k}](\bigvee_{f \stackrel{\alpha,k}{\longrightarrow} f'} \varrho_{f',k})$. Suppose the converse, i.e., $g \models \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle (\bigwedge_{f \stackrel{\alpha,k}{\longrightarrow} f'} \neg \varrho_{f',k})$. By applying the induction hypothesis we obtain that there is $g \stackrel{\alpha,k}{\rightarrow} g'$ such that for every $f \stackrel{\alpha,k}{\rightarrow} f'$ we have $g' \not\models \varrho_{f',k}$, i.e., $(g', f') \notin T(\leqslant_k)$. Hence, $g \not\leqslant_{k+1} f$ which is a contradiction.

"⇒": As $g \models \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$, for every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \doteqdot_k f'$ (by induction hypothesis). We show that $g \leqslant_{k+1} f$. Let $g \stackrel{\alpha,k}{\rightarrow} g'$. Since g, g' are reachable from g, there are $f_1, f_2 \in F$ such that $g \doteqdot_k f_1$ and $g' \doteqdot_k f_2$. Since $g \models [\alpha, \varphi_{f_1,k}, \psi_{f_2,k}](\bigvee_{f \stackrel{\alpha,k}{\longrightarrow} f'} \varrho_{f',k})$, we have that $g' \models \bigvee_{f \stackrel{\alpha,k}{\longrightarrow} f'} \varrho_{f',k}$ by using the induction hypothesis. Hence, there is $f \stackrel{\alpha,k}{\rightarrow} f'$ such that $g' \models \varrho_{f',k}$, which means $(g', f') \in T(\leqslant_k)$ (again by induction hypothesis). □

In general, the $\leqslant_k$-consistency of moves $g \stackrel{\alpha}{\Rightarrow} g'$ can be expressed in a given logic only if one can express the $\doteqdot_k$ equivalence with g and g'. Since g and g' can be infinite-state processes, this is generally impossible. This difficulty was overcome in Theorem 14 by using the assumption that g and g' are $\doteqdot_k$ equivalent to some f_1 and f_2 of F. Thus, we only needed to encode the $\doteqdot_k$ 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 9 and Theorem 14 is the following:

Corollary 15 *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) $g \models \Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n^2})$.

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).

Remark 16 *An important consequence of Corollary 15 is that the problem of full regular equivalence is generally "more decidable and tractable" than the* *problem of regular equivalence. For example, regular weak simulation equivalence for PA, PAN, and lossy channel systems is undecidable[23], while model-checking with the logic $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ (and thus also the problem of full regular MTB equivalence) is still decidable for these models [27,25]. Another example are pushdown processes. Model-checking $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ for PDA is in **PSPACE** [39]. As we shall see, this means that the full regular MTB equivalence problem for PDA is also in **PSPACE**. However, the regular weak simulation equivalence problem for PDA is **EXPTIME**-complete [22]. Further examples are given below. Hence, the "extra" reachability condition given in the definition of full regular equivalence problem is a crucial ingredient of our result, and not just a handy technical assumption which could possibly be avoided.*

A natural question is what is the complexity of the reduction from (a) to (b) in Corollary 15. At first glance, it seems to be exponential because the size of Θ_{f',n^2} is exponential in the size of $\mathcal{F}$. However, the number of distinct subformulae in Θ_{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* [6], 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 Lemma 18. We start by proving an auxiliary technical lemma:

Lemma 17 *For every $k \in \mathbb{N}_0$, the relation $\xrightarrow{\alpha,k+1}$ over $F \times F$ can be computed in $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time, assuming that the relation $\leqslant_k$ over $F \times F$ has already been computed.*

PROOF. We assume that binary relations are stored as bit matrices, which means that testing the membership to $\leqslant_k$ for a given pair of processes $f_1, f_2 \in F$ can be done in constant time.

First we show how to compute $\xrightarrow{\alpha,k}$ from $\leqslant_k$ in $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time. This is easy—for every $\alpha \in \mathcal{A}$ we examine $\mathcal{O}(n^2)$ pairs $f_1, f_2 \in F$ and decide if $f_1 \xrightarrow{\alpha,k} f_2$. Since testing the membership to $\leqslant_k$ is for free, this is not harder than reachability which can be done in $\mathcal{O}(n^2)$ time. Hence, we need $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time in total.

Now we show that $\leqslant_{k+1}$ can be computed from $\xrightarrow{\alpha,k}$ and $\leqslant_k$ in $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time. By definition of $\leqslant_{k+1}$, we need to examine $\mathcal{O}(n^2)$ pairs $f_1, f_2 \in F$ and for each of $\mathcal{O}(n \cdot |\mathcal{A}|)$ moves $f_1 \xrightarrow{\alpha,k} f_1'$ we check $\mathcal{O}(n)$ possible responses $f_2 \xrightarrow{\alpha,k} f_2'$ and

[6] A circuit (or a DAG) representing a formula φ is basically the syntax tree for φ where the nodes representing the same subformula are identified.

look if $(f_1, f_2) \in T(\leqslant_k)$ (the membership to $T(\leqslant_k)$ is also for free if $\leqslant_k$ is stored as a bit matrix). Hence, $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time suffices.

Now $\xrightarrow{\alpha,k+1}$ is computed from $\leqslant_{k+1}$ as above (i.e., in $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time) and we are done. □

Lemma 18 *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. (Here $\mathcal{P}$ is the polynomial introduced in Definition 10.)*

PROOF. We show that for every $k \in \mathbb{N}_0$, one only needs $\mathcal{O}(n^4 \cdot |\mathcal{A}| \cdot k + \mathcal{P}(n, |\mathcal{A}|))$ time to compute

- the relation $\leqslant_k$ over $F \times F$, and
- a circuit such that all $\Phi_{f,k}$, $\Psi_{f,k}$, and $\Theta_{f,k}$, where $f \in F$, are represented by some nodes of the circuit.

We proceed by induction on k. The case when $k = 0$ follows immediately—we just compute $\leqslant_0$ over $F \times F$ and the circuits for all Ξ_f. This takes $\mathcal{P}(n, |\mathcal{A}|)$ time. In the inductive step we first compute $\xrightarrow{\alpha,k+1}$ and $\leqslant_{k+1}$ over $F \times F$. This can be done in $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time, because the relation $\leqslant_k$ has been computed in the previous step and hence we can apply Lemma 17. Now observe that if we already have a circuit representing all $\Phi_{f,k}$, $\Psi_{f,k}$ and $\Theta_{f,k}$, then we need to add only $\mathcal{O}(n^3 \cdot |\mathcal{A}|)$ new nodes to obtain a circuit representing $\Phi_{\bar{f},k+1}$ for a *given* $\bar{f} \in F$, and this procedure does not take more than $\mathcal{O}(n^3 \cdot |\mathcal{A}|)$ time. This follows immediately from the definition of $\Phi_{\bar{f},k+1}$ and the fact that the problem if $f_1 \xrightarrow{\alpha,k+1} f_2$ for given $f_1, f_2 \in F$ can now be decided in constant time (because we have computed $\xrightarrow{\alpha,k+1}$ over $F \times F$). The same actually holds for the formula $\Psi_{\bar{f},k+1}$. Hence, we only add $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ new nodes in $\mathcal{O}(n^4 \cdot |\mathcal{A}|)$ time to obtain a circuit representing all $\Phi_{f,k+1}$, $\Psi_{f,k+1}$, and $\Theta_{f,k+1}$. By applying the induction hypothesis, we obtain that $\mathcal{O}(n^4 \cdot |\mathcal{A}| \cdot (k+1) + \mathcal{P}(n, |\mathcal{A}|))$ time suffices to compute $\leqslant_{k+1}$ and the circuit representing all $\Phi_{f,k+1}$, $\Psi_{f,k+1}$, and $\Theta_{f,k+1}$. □

Corollary 15 and Lemma 18 can also be applied to finite-state processes (i.e., to processes of some finite-state system $\mathcal{F}$).

Corollary 19 *Let $\sim$ be an MTB equivalence where B is well-defined. The problem of checking $\sim$ between finite-state processes is efficiently reducible to the model checking problems with the logics $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ and $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$ over finite-state processes.*

The previous corollary is actually interesting only for those MTB equivalences where $M = \emptyset$, because otherwise we must compute the $\leqslant_{n^2} = \leqslant$ relation over $F \times F$ just to construct the formula given in Corollary 15 (b). If $M = \emptyset$, there is no need to construct the $\leqslant_k$ relations, because $\stackrel{\alpha,k}{\rightarrow} = \stackrel{\alpha}{\Rightarrow}$ for every $k \in \mathbb{N}_0$. Hence, the construction of the formula of Corollary 15 (b) is rather simple in this case. Thus, one might re-use existing model-checking tools for finite-state processes to experiment with MTB equivalences over finite-state processes.

4 PS Preorder and Equivalence

In this section we consider another parameterized family of process preorders/equivalences whose definitions are based on inclusion/equality of sets of decorated traces.

Definition 20 *Let P be a process preorder and $S \in \{\gamma, \lambda\}$ a* scope *(here γ and λ stands for "global" and "local", respectively). For each $k \in \mathbb{N}_0$ we define the relation $\sqsubseteq_k$ over processes as follows: $s \sqsubseteq_k t$ iff for every sequence $s = s_0 \stackrel{\alpha_1}{\Rightarrow} s_1 \stackrel{\alpha_2}{\Rightarrow} \cdots \stackrel{\alpha_i}{\Rightarrow} s_i$, where $0 \le i \le k$, there is a matching sequence $t = t_0 \stackrel{\alpha_1}{\Rightarrow} t_1 \stackrel{\alpha_2}{\Rightarrow} \cdots \stackrel{\alpha_i}{\Rightarrow} t_i$ such that*

- *if $S = \lambda$, then $(s_i, t_i) \in P$;*
- *if $S = \gamma$, then $(s_j, t_j) \in P$ for all $0 \le j \le i$.*

PS preorder*, denoted $\sqsubseteq$, is defined by $s \sqsubseteq t$ iff $s \sqsubseteq_k t$ for every $k \in \mathbb{N}_0$.* PS equivalence*, denoted $\sim$, is defined by $s \sim t$ iff $s \sqsubseteq t$ and $t \sqsubseteq s$.*

For example, let us consider the preorders T, D, F, R, U defined as follows (where $I(s) = \{a \in Act \mid s \stackrel{a}{\Rightarrow} t \text{ for some } t\}$):

- $(s, t) \in T$ 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).
- $(s, t) \in U$ iff s and t are trace equivalent (that is, iff $\{w \in Act^* \mid \exists s \stackrel{w}{\Rightarrow} s'\} = \{w \in Act^* \mid \exists t \stackrel{w}{\Rightarrow} t'\}$).

Now one can readily check that $T\lambda$, $D\lambda$, $F\lambda$, $F\gamma$, $R\lambda$, $R\gamma$, and $U\lambda$ 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.

For the rest of this section, let us fix a process preorder P and a scope S. Now we give another characterization of PS preorder/equivalence which is more convenient for our purposes.

Let M, N be sets of processes. We write $M \stackrel{\alpha}{\Rightarrow} N$ iff for every $t \in N$ there is some $s \in M$ such that $s \stackrel{\alpha}{\Rightarrow} t$.

Definition 21 *For every $i \in \mathbb{N}_0$ we inductively define the relation $\preccurlyeq_i$ between processes and non-empty sets of processes as follows:*

- *$s \preccurlyeq_0 M$ for every process s and every non-empty set of processes M such that*
  - *if $S = \gamma$, then $(s,t) \in P$ for every $t \in M$;*
  - *if $S = \lambda$, then $(s,t) \in P$ for some $t \in M$.*
- *$s \preccurlyeq_{i+1} M$ iff $s \preccurlyeq_i M$ and for every $s \stackrel{\alpha}{\Rightarrow} t$ there is $M \stackrel{\alpha}{\Rightarrow} N$ such that $t \preccurlyeq_i N$.*

We put $s \preccurlyeq M$ iff $s \preccurlyeq_i M$ for every $i \in \mathbb{N}_0$. Slightly abusing notation, we write $s \preccurlyeq_i t$ and $s \preccurlyeq t$ instead of $s \preccurlyeq_i \{t\}$ and $s \preccurlyeq \{t\}$, respectively. We also write $s \approx t$ iff $s \preccurlyeq t$ and $t \preccurlyeq s$.

Lemma 22 *For every $i \in \mathbb{N}_0$ and all processes s, t we have that $s \sqsubseteq_i t$ iff $s \preccurlyeq_i t$ (hence, $s \sqsubseteq t$ iff $s \preccurlyeq t$, and $s \sim t$ iff $s \approx t$).*

PROOF. First, let us extend the $\sqsubseteq_k$ relations so that they also relate processes to non-empty sets of processes—by writing $s \sqsubseteq_k M$ we mean that for every sequence $s = s_0 \stackrel{\alpha_1}{\Rightarrow} s_1 \stackrel{\alpha_2}{\Rightarrow} \cdots \stackrel{\alpha_i}{\Rightarrow} s_i$, where $0 \leq i \leq k$, there is $t \in M$ and a sequence $t = t_0 \stackrel{\alpha_1}{\Rightarrow} t_1 \stackrel{\alpha_2}{\Rightarrow} \cdots \stackrel{\alpha_i}{\Rightarrow} t_i$ such that

- if $S = \lambda$, then $(s_i, t_i) \in P$;
- if $S = \gamma$, then $(s_j, t_j) \in P$ for all $0 \leq j \leq i$.

We also require that M is *minimal* in the sense that each non-empty proper subset of M violates the condition above.

Note that $s \sqsubseteq_i t$ iff $s \sqsubseteq_i \{t\}$. Also note that if $s \sqsubseteq_i M$, then there is some $t \in M$ such that $(s,t) \in P$; and if $S = \gamma$, then $(s,t) \in P$ for *every* $t \in M$ (this is where we need the minimality of M). Now we prove that $s \sqsubseteq_i M$ iff $s \preccurlyeq_i M$ for every process s and every non-empty set of processes M. We proceed by induction on i. The case when $i = 0$ follows directly from definitions. Now suppose that $s \sqsubseteq_{i+1} M$. We need to show that also $s \preccurlyeq_{i+1} M$, i.e., for each $s \stackrel{\alpha}{\Rightarrow} t$ there is some $M \stackrel{\alpha}{\Rightarrow} N$ such that $t \preccurlyeq_i N$. Since $s \sqsubseteq_{i+1} M$ and $s \stackrel{\alpha}{\Rightarrow} t$, there must be a minimal subset N consisting of (some) $\stackrel{\alpha}{\Rightarrow}$ successors of states in M such that $t \sqsubseteq_i N$. Hence, $M \stackrel{\alpha}{\Rightarrow} N$, and $t \preccurlyeq_i N$ by induction hypothesis. Similarly, one can also show that if $s \preccurlyeq_{i+1} M$, then $s \sqsubseteq_{i+1} M$. $\square$

Due to Lemma 22 we can safely consider the relations $\preccurlyeq_i$, $\preccurlyeq$, and $\approx$ instead of the relations $\sqsubseteq_i$, $\sqsubseteq$, and $\sim$, respectively. The next two lemmas are immediate:

Lemma 23 *Let $\mathcal{F} = (F, \rightarrow, \mathcal{A})$ be a finite-state system with n states. Then $\preccurlyeq_{n2^n-1} \;=\; \preccurlyeq_{n2^n} \;=\; \preccurlyeq$, where all of the relations are considered as being restricted to $F \times 2^F$.*

Lemma 24 *For all processes s, t, and non-empty sets of processes M, N, and every $i \in \mathbb{N}_0$ we have that*

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

Now we can state and prove the crucial theorem:

Theorem 25 *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) $g \approx f$ and for every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \approx f'$.
(b) $g \approx_{n2^n} f$ and for every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \approx_{n2^n} f'$.

PROOF. $(a) \Rightarrow (b)$ is immediate. For the other direction, suppose that (b) holds and (a) does not hold. Since (a) does not hold, there is $g \rightarrow^* g'$ such that $g' \not\approx f'$ for every $f' \in F$; and as (b) holds, there is some $\bar{f} \in F$ such that $g' \approx_{n2^n} \bar{f}$. To sum up, we have that $g' \not\approx_m \bar{f}$ for some $m > n2^n$. Now we distinguish two possibilities:

$g' \not\preccurlyeq_m \bar{f}$**.** By definition of $\preccurlyeq_i$ (and the fact that $m > n2^n$), there must be some $g' \rightarrow^* g''$ and $M \subseteq F$ such that $g'' \preccurlyeq_{n2^n-1} M$ and $g'' \not\preccurlyeq_{n2^n} M$. We show that this is impossible. To see this, realize that $g \rightarrow^* g''$ and due to (b) there is some $f' \in F$ such that $g'' \approx_{n2^n} f'$. So, $f' \preccurlyeq_{n2^n} g'' \preccurlyeq_{n2^n-1} M$, which means $f' \preccurlyeq_{n2^n-1} M$ by Lemma 24 (a). Hence, $f' \preccurlyeq_{n2^n} M$ by Lemma 23. Now $g'' \preccurlyeq_{n2^n} f' \preccurlyeq_{2^n} M$ and thus we obtain $g'' \preccurlyeq_{n2^n} M$ by applying Lemma 24 (a), which is a contradiction.

$\bar{f} \not\preccurlyeq_m g'$**.** Then there must be some $\bar{f} \rightarrow^* f'$ and a set of processes M such that every $g'' \in M$ is reachable from g', $f' \preccurlyeq_{n2^n-1} M$, and $f' \not\preccurlyeq_{n2^n} M$. Again, this will be led to a contradiction. Since every process of M is reachable from g, due to (b) there is a set $N \subseteq F$ such that for every $g'' \in M$ there is $f'' \in N$ such that $g'' \approx_{n2^n} f''$, and vice versa. Hence, $f' \preccurlyeq_{n2^n-1} N$ by Lemma 24 (b), which means that $f' \preccurlyeq_{n2^n} N$ by Lemma 23. Thus, we obtain $f' \preccurlyeq_{n2^n} M$ again by applying Lemma 24 (b) (the roles of M, N are interchanged now), which is a contradiction. $\square$

Now we show how to encode the condition (b) of Theorem 25 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, $[\![\alpha]\!]\varphi \equiv \neg\langle\!\langle\alpha\rangle\!\rangle\neg\varphi$. 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 26 *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 Ξ_f, Γ_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 U preorder is (provably) not well-defined. Nevertheless, our results *do* apply to possible-futures equivalence, as we shall see in Remark 31.

Lemma 27 *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 PS Preorder into Modal Logic

Definition 28 *For all $i \in \mathbb{N}_0$, $f \in F$, and $M \subseteq F$ we define the sets*

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

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:

- $\Phi_{f,0} = \Xi_f,\ \Psi_{f,0} = \Gamma_f$
- $\Theta_{f,k} = \Phi_{f,k} \wedge \Psi_{f,k}$
- $\Phi_{f,k+1} = \Xi_f \ \wedge\ \Big(\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}\Big) \ \wedge \bigwedge_{f \stackrel{\alpha}{\Rightarrow} f'} \ \bigvee_{M \in \mathcal{F}(f', \preccurlyeq_k)} \ \bigwedge_{f'' \in M} \langle\!\langle\alpha\rangle\!\rangle\Theta_{f'',k}$
- $\Psi_{f,k+1} = \Gamma_f \ \wedge\ \Big(\mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}\Big) \ \wedge \bigwedge_{\alpha \in \mathcal{A}_\tau} [\![\alpha]\!]\Big(\bigvee_{f \stackrel{\alpha}{\Rightarrow} M} \ \bigvee_{f' \in \mathcal{F}(\preccurlyeq_k, M)} \Theta_{f',k}\Big)$

The empty conjunction is equivalent to `tt`*, and the empty disjunction to* `ff`*.*

The $\mathcal{F}(\ldots)$ sets are effectively constructible in time exponential in n and polynomial in i (Lemma 27), hence the $\Phi_{f,k}$, $\ldots$, formulae are effectively constructible too.

Theorem 29 *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 $f \preccurlyeq_0 g$; further, $g \models \Phi_{f,k+1}$ iff $f \preccurlyeq_{k+1} g$ and for each $g \to^* g'$ there is $f' \in F$ such that $g' \approx_k f'$.
(b) $g \models \Psi_{f,0}$ iff $g \preccurlyeq_0 f$; further, $g \models \Psi_{f,k+1}$ iff $g \preccurlyeq_{k+1} f$ and for each $g \to^* g'$ there is $f' \in F$ such that $g' \approx_k f'$.
(c) $g \models \Theta_{f,0}$ iff $g \approx_0 f$; further, $g \models \Theta_{f,k+1}$ iff $g \approx_{k+1} f$ and for each $g \to^* g'$ there is $f' \in F$ such that $g' \approx_k f'$.

PROOF. The (a), (b), and (c) are proved simultaneously by induction on k. We give explicit arguments just for (a) and (b); the (c) follows immediately then.

- $k = 0$. Immediate.
- **Induction step.**
  "(a), $\Rightarrow$" Let $g \models \Phi_{f,k+1}$. Then $g \models \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ and hence for every $g \to^* g'$ there is some $f' \in F$ such that $g' \approx_k f'$ by applying the induction hypothesis. We show that $f \preccurlyeq_{k+1} g$. As $g \models \Xi_f$, we have that $(f, g) \in P$. Let $f \stackrel{\alpha}{\Rightarrow} f'$. Since $g \models \bigwedge_{f \stackrel{\alpha}{\Rightarrow} f'} (\bigvee_{M \in \mathcal{F}(f', \preccurlyeq_k)} (\bigwedge_{f'' \in M} \langle\!\langle \alpha \rangle\!\rangle \Theta_{f'',k}))$, there is $M \subseteq F$ such that $f' \preccurlyeq_k M$ (this follows from the definition of $\mathcal{F}(f', \preccurlyeq_k)$). Let $M = \{f_1, \ldots, f_m\}$. As $g \models \bigwedge_{f'' \in M} \langle\!\langle \alpha \rangle\!\rangle \Theta_{f'',k}$, we can use the induction hypothesis to conclude that there is a set $N = \{g_1, \cdots, g_m\}$ where for every $0 \leq i \leq m$ we have that $g \stackrel{\alpha}{\Rightarrow} g_i$ and $g_i \approx_k f_i$. Note that $g \stackrel{\alpha}{\Rightarrow} N$. We claim that $f' \preccurlyeq_k N$. However, this follows immediately from Lemma 24 (b).

  "(a), $\Leftarrow$" Let us assume that $f \preccurlyeq_{k+1} g$ and for every $g \to^* g'$ there is $f' \in F$ such that $g' \approx_k f'$. Then $g \models \Xi_f \wedge \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ by applying the definition of $\preccurlyeq_{k+1}$ and the induction hypothesis. Since $f \preccurlyeq_{k+1} g$, for every $f \stackrel{\alpha}{\Rightarrow} f'$ there is some $g \stackrel{\alpha}{\Rightarrow} N$ such that $f' \preccurlyeq_k N$. Now let $M = \{f'' \in F \mid f'' \approx_k g'' \text{ for some } g'' \in N\}$. Since every state of N is reachable from g, for every $g'' \in N$ there is at least one $f'' \in M$ such that $g'' \approx_k f''$. As $f' \preccurlyeq_k N$, we also have that $f' \preccurlyeq_k M$ by applying Lemma 24 (b). Hence, $M \in \mathcal{F}(f', \preccurlyeq_k)$. To sum up, we obtain that $g \models \bigwedge_{f \stackrel{\alpha}{\Rightarrow} f'} (\bigvee_{M \in \mathcal{F}(f', \preccurlyeq_k)} (\bigwedge_{f'' \in M} \langle\!\langle \alpha \rangle\!\rangle \Theta_{f'',k}))$ and we are done.

  "(b), $\Rightarrow$" Let $g \models \Psi_{f,k+1}$. Then $g \models \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ and hence for every $g \to^* g'$ there is some $f' \in F$ such that $g' \approx_k f'$ by applying the induction hypothesis. We show that $g \preccurlyeq_{k+1} f$. As $g \models \Gamma_f$, we have that $(g, f) \in P$. Let $g \stackrel{\alpha}{\Rightarrow} g'$. Since $g \models \bigwedge_{\alpha \in \mathcal{A}_\tau} [\![\alpha]\!] (\bigvee_{f \stackrel{\alpha}{\Rightarrow} M} \bigvee_{f' \in \mathcal{F}(\preccurlyeq_k, M)} \Theta_{f',k})$, there are $f \stackrel{\alpha}{\Rightarrow} M$ and $f' \in F$ such that $f' \preccurlyeq_k M$ and $g' \approx_k f'$ (here we apply the definition of

$\mathcal{F}(\preccurlyeq_k, M)$ and the induction hypothesis). Since $g' \preccurlyeq_k f' \preccurlyeq_k M$, we obtain $g' \preccurlyeq_k M$ by Lemma 24 (a).

"(b), $\Leftarrow$" Let us assume that $g \preccurlyeq_{k+1} f$ and for every $g \rightarrow^* g'$ there is $f' \in F$ such that $g' \approx_k f'$. Then $g \models \Gamma_f \wedge \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ by applying the definition of $\preccurlyeq_{k+1}$ and the induction hypothesis. Since $g \preccurlyeq_{k+1} f$, for every $g \stackrel{\alpha}{\Rightarrow} g'$ there is some $f \stackrel{\alpha}{\Rightarrow} M$ such that $g' \preccurlyeq_k M$. Further, as g' is reachable from g, there is some $f' \in F$ such that $g' \approx_k f'$. Since $f' \preccurlyeq_k g' \preccurlyeq_k M$, we obtain $f' \preccurlyeq_k M$ by Lemma 24 (a). This means that $f' \in \mathcal{F}(\preccurlyeq_k, M)$. To sum up, we have that $g \models \bigwedge_{\alpha \in \mathcal{A}_\tau} [\![\alpha]\!] (\bigvee_{f \stackrel{\alpha}{\Rightarrow} M} \bigvee_{f' \in \mathcal{F}(\preccurlyeq_k, M)} \Theta_{f',k})$ and the proof is finished. $\square$

Corollary 30 *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) $g \approx f$ and for every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \approx f'$.
(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 31 *As we already mentioned, the U 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 preorder is not well-defined.*

5 Model Checking Lossy Channel Systems

In this section we show that the model checking of $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}, \mathbf{EU}_\alpha)$ 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.

Definition 32 *[1] A* channel system *is a tuple $S = (Q, C, \Sigma, \mathcal{A}, \Delta)$, where Q is a finite set* control states, *$C = \{c_1, \cdots, c_k\}$ is a finite set of* channels, *$\mathcal{A}$ is a finite alphabet of* actions, *Σ is a finite alphabet of* messages, *and Δ is a finite set of* transition rules, *each of which is a triple of the form (q, op, q'), where $q, q' \in Q$ and op is an* operation *of one of the forms*

- *$c!u$, where $c \in C$ and $u \in \Sigma$,*
- *$c?u$, where $c \in C$ and $u \in \Sigma$,*
- *$\alpha \in \mathcal{A}_\tau$.*

A *configuration* of S is a tuple $\sigma = \langle q, w_1, \cdots, w_k\rangle$, where $q \in Q$ is a control state and $w_1, \cdots, w_k \in \Sigma^*$ are words denoting the sequences of messages stored in channels. The transition rules in Δ state how S can move from one configuration to another. Formally, S has a "perfect" step $\sigma \xrightarrow{\alpha}_p \sigma'$ iff σ is some $\langle q, w_1, \cdots, w_k\rangle$, σ' is some $\langle q', w_1, \cdots, w_{i-1}, v, w_{i+1}, \cdots, w_k\rangle$, and one of the following conditions is satisfied:

- $\alpha = \tau$ and there is a rule $(q, c_i!u, q') \in \Delta$ such that $v = w_i u$ (i.e., u has been written to c_i);
- $\alpha = \tau$ and there is a rule $(q, c_i?u, q') \in \Delta$ such that $w_i = uv$ (i.e., u has been read from c_i);
- $v = w_i$ and there is a rule $(q, \alpha, q') \in \Delta$ (i.e., the action α has been performed without changing the contents of channels).

These steps are called perfect because no messages are lost. Assuming perfect steps, channel systems (even systems with just one channel) can faithfully simulate an arbitrary Turing machine with quadratic overhead [7]. Hence all non-trivial verification problems are undecidable for LCS's.

Saying that a channel system is *lossy* means that messages can be lost while they are in the channels. This is formally captured by introducing an ordering between configurations. We write $u \sqsubseteq v$ if u is a "scattered subword" of v, i.e., if one can obtain u by erasing some letters in v (possibly all letters, possibly none). This ordering is extended to configurations as follows: $\langle q, w_1, \cdots, w_k\rangle \leq \langle q', w'_1, \cdots, w'_k\rangle$ when $q = q'$ and $w_i \sqsubseteq w'_i$ for all $1 \leq i \leq k$. By Higman's lemma, $\leq$ is a well-quasi-ordering (a *wqo*), i.e., it is well-founded and every set of incomparable configurations is finite.

Now the *lossy steps* of a given channel system S are defined as follows: $\sigma \xrightarrow{\alpha} \sigma'$ iff either $\theta \xrightarrow{\alpha}_p \theta'$ for some configurations θ, θ' such that $\sigma \geq \theta$ and $\theta' \geq \sigma'$, or $\alpha = \tau$ and $\sigma \neq \sigma' \leq \sigma$. Note that $\sigma_1 \geq \sigma_2 \xrightarrow{\alpha} \sigma_3 \geq \sigma_4$ entails $\sigma_1 \xrightarrow{\alpha} \sigma_4$. The transition system associated with a LCS $S = (Q, C, \Sigma, \mathcal{A}, \Delta)$ as above is $T = (Q \times \Sigma^{*k}, \rightarrow, \mathcal{A})$, where the lossy steps are taken into account.

We are interested in sets of configurations denoted by some simple expressions. For a configuration σ we let $\uparrow\sigma$ denote the upward-closure of σ, i.e., the set $\{\theta \mid \sigma \leq \theta\}$. A *restricted set* is denoted by an expression ϱ of the form

$$\uparrow\sigma - \uparrow\theta_1 - \ldots - \uparrow\theta_n$$

where $\sigma, \theta_1, \ldots, \theta_n$ are some configurations. This denotes the set $\uparrow\sigma$ minus the "restrictions" $\uparrow\theta_i$.

An expression ϱ is *trivial* if it denotes the empty set. Clearly $\uparrow\sigma - \uparrow\theta_1 - \cdots - \uparrow\theta_n$ is trivial iff $\theta_i \leq \sigma$ for some i. A *constrained set* is a finite union of restricted sets, denoted by an expression γ of the form $\varrho_1 \vee \cdots \vee \varrho_m$ (if $m = 0$, i.e., when the disjunction is empty, we may write just *false*). Such an expression is *reduced* if no ϱ_i is trivial and it is easy to transform any constrained set in an equivalent reduced one. For a set M of configurations and $\alpha \in \mathcal{A}_\tau$, let $Pre_\alpha(M) = \{\sigma \mid \sigma \xrightarrow{\alpha} \sigma' \text{ for some } \sigma' \in M\}$ be the set of all immediate α-predecessors of configurations in M.

In the rest of this section we do not strictly distinguish between sets of configurations and expressions denoting these sets. For example, if ϱ is an expression denoting a constrained set M, we write $Pre_\alpha(\varrho)$ instead of $Pre_\alpha(M)$.

Now we show that constrained sets are closed under Boolean operations, and that expressions like $\gamma_1 \wedge \gamma_2$ or $\neg\gamma$ can effectively be transformed into equivalent reduced expressions. Additionally, constrained sets are effectively closed under Pre_α. These results enable symbolic model-checking of $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}, \mathbf{EU}_\alpha)$ formulae for LCS's, where reduced expressions are used to represent sets of configurations that satisfy individual subformulae. For the rest of this section, let us fix a channel system $S = (Q, C, \Sigma, \mathcal{A}, \Delta)$ where $C = \{c_1, \cdots, c_k\}$.

Lemma 33 *Constrained sets are closed under intersection. Furthermore, from reduced expressions γ_1 and γ_2, one can compute a reduced expression for $\gamma_1 \wedge \gamma_2$.*

PROOF. For all $v, w \in \Sigma^*$, let $v \| w$ be the set consisting of all $u \in \Sigma^*$ such that

- $v \sqsubseteq u$, $w \sqsubseteq u$,
- for every word $u' \neq u$ such that $v \sqsubseteq u'$, $w \sqsubseteq u'$ we have that $u' \not\sqsubseteq u$.

In other words, $u \in v\|w$ iff u is a minimal upper bound of $\{v, w\}$ w.r.t. $\sqsubseteq$. For example, $aba\|cab = \{caba, abcab, abcba\}$. Note that $|u| \leq |v| + |w|$. Hence, the set $v\|w$ is finite and effectively computable (e.g., by exhaustive search).

The intersection $\uparrow\langle q, w_1, \cdots, w_k\rangle \wedge \uparrow\langle q', w_1', \cdots, w_k'\rangle$ of two upward-closures is empty when $q \neq q'$. Otherwise, it is equal to

$$\bigvee_{u_1 \in w_1\|w_1'} \cdots \bigvee_{u_k \in w_k\|w_k'} \uparrow\langle q, u_1, \cdots, u_k\rangle$$

For example, $\uparrow\langle q, aba\rangle \wedge \uparrow\langle q, cab\rangle = \uparrow\langle q, caba\rangle \vee \uparrow\langle q, abcab\rangle \vee \uparrow\langle q, abcba\rangle$.

The intersection of restricted sets follows easily—assuming

$$\uparrow\sigma \wedge \uparrow\sigma' = \uparrow\sigma_1 \vee \ldots \vee \uparrow\sigma_\ell,$$

one derives

$$(\uparrow\sigma - \uparrow\theta_1 - \ldots - \uparrow\theta_n) \wedge (\uparrow\sigma' - \uparrow\theta_{n+1} - \ldots - \uparrow\theta_m) = \bigvee_{i=1}^{m} \Big(\uparrow\sigma_i - \uparrow\theta_1 - \ldots - \uparrow\theta_m\Big). \quad (1)$$

This allows intersecting constrained sets: $(\bigvee_i \varrho_i) \wedge (\bigvee_j \varrho_j) = \bigvee_i \bigvee_j (\varrho_i \wedge \varrho_j)$. $\square$

Lemma 34 *Constrained sets are closed under complementation. Furthermore, from a reduced expression γ, one can compute a reduced expression for $\neg\gamma$.*

PROOF. Complementation is easy for upward-closures:

$$\neg\uparrow\langle q, w_1, \cdots, w_k\rangle = \Big(\uparrow\langle q, \varepsilon, \cdots, \varepsilon\rangle - \uparrow\langle q, w_1, \cdots, w_k\rangle\Big) \ \vee \ \bigvee_{q' \neq q} \uparrow\langle q', \varepsilon, \cdots, \varepsilon\rangle.$$

This allows complementing restricted sets:

$$\neg(\uparrow\sigma - \uparrow\theta_1 - \ldots - \uparrow\theta_n) = \uparrow\theta_1 \vee \cdots \vee \uparrow\theta_n \vee \neg\uparrow\sigma.$$

We use intersection (Lemma 33) for complementing constrained sets:

$$\neg(\varrho_1 \vee \cdots \vee \varrho_m) = (\neg\varrho_1) \wedge \cdots \wedge (\neg\varrho_m). \qquad \square$$

Lemma 35 *Constrained sets are closed under Pre_α. Furthermore, from a reduced expression γ, one can compute a reduced expression for $Pre_\alpha(\gamma)$.*

PROOF. Since $Pre_\alpha(\bigvee_i \varrho_i) = \bigvee_i Pre_\alpha(\varrho_i)$, it is enough to compute $Pre_\alpha(\varrho)$ for ϱ a restricted set. If ϱ has the reduced form $\uparrow\sigma - \uparrow\theta_1 - \ldots - \uparrow\theta_n$, then $Pre_\alpha(\varrho) = Pre_\alpha(\uparrow\sigma) = Pre_\alpha(\sigma)$ (cf. the definition of lossy steps). We assume $\sigma = \langle q, w_1, \ldots, w_k\rangle$ and show how to express $Pre_\alpha(\sigma)$ as a finite union of upward closures. There are two cases:

$\alpha \neq \tau$: Let Γ contains all configurations $\langle q', w_1, \ldots, w_n\rangle$ for $q' \in Q$ such that $(q', \alpha, q) \in \Delta$.

$\alpha = \tau$: Here constructing Γ is a bit more involved. For all rules $(q', c_i?u, q) \in \Delta$, we put $\langle q', w_1, \ldots, uw_i, \ldots, w_k\rangle$ in Γ. For all rules $(q', c_i!u, q) \in \Delta$, the configuration we put in Γ is $\langle q', w_1, \ldots, w_i, \ldots, w_k\rangle$ if w_i does not end with u, or $\langle q', w_1, \ldots, v, \ldots, w_k\rangle$ if $w_i = vu$. Finally, we put in Γ all configurations $\sigma' > \sigma$ s.t. σ' differs from σ by just one message.

In both cases, $Pre_\alpha(\sigma) = \bigvee_{\theta\in\Gamma} \uparrow\theta$. We conclude by noting that Γ is finite and effectively constructible. $\square$

We can now compute the set of configurations that satisfy an $\mathbf{EU}_\alpha$ formula:

Lemma 36 *Let S_1 and S_2 be two constrained sets. Then the set S of configurations that satisfy $S_1\,\mathbf{EU}_\alpha\,S_2$ is constrained too. Furthermore, from reduced expressions for S_1 and S_2, one can compute a reduced expression for S.*

PROOF. We inductively define a sequence $(U_i)_{i\in\mathbb{N}_0}$ of sets of configurations as follows:

- $U_0 = \begin{cases} \mathit{Pre}_\alpha(S_2) & \text{if } \alpha \neq \tau \\ S_2 & \text{otherwise.} \end{cases}$
- $U_{i+1} = U_i \cup (\mathit{Pre}_\tau(U_i) \cap S_1)$

Then $S = \bigcup_{i\in\mathbb{N}_0} U_i$.

By the previous Lemmas, every U_i is a constrained set and one can compute, for each $S_1 \cap \mathit{Pre}(U_i)$, a reduced expression $\bigvee_j \varrho_{i,j}$ with $\varrho_{i,j}$ having the form $\uparrow\sigma_{i,j} - \uparrow\theta_{i,j,1} - \ldots - \uparrow\theta_{i,j,\ell}$. The crucial point in our proof is that *all restrictions $\theta_{i,j,k}$ already occur in the expression for S_1*. Indeed, the algorithm for Pre_α (Lemma 35) does not use restrictions, and the algorithm for intersection (see Equation (1) in Lemma 33) only uses restrictions that were already present.

Assume now that the sequence of U_i's is strictly increasing. Then for every i there is some j_i such that ϱ_{i,j_i} is not included in U_i. Extract from the sequence $(\varrho_{i,j_i})_i$ an infinite subsequence where the restrictions are always the same (this can be done since the restrictions come from a finite set). Now the wqo property of $\leq$ entails that some ϱ_{i,j_i} in this sequence is included in a previous $\varrho_{i',j_{i'}}$, contradicting the assumption that ϱ_{i,j_i} is not included in U_i, a superset of $U_{i'+1}$.

Hence, the sequence of U_i's eventually stabilize. Since it is possible to compare U_{i+1} with U_i when we compute it, stabilization can be detected. At stabilization, we have computed a reduced expression for S. $\square$

The **EU** operator can be handled similarly as $\mathbf{EU}_\alpha$. We just treat all actions as if they were τ and use the algorithm for $\mathbf{EU}_\tau$. Thus, we obtain the following:

Lemma 37 *Let S_1 and S_2 be two constrained sets. Then the set S of configurations that satisfy $S_1\,\mathbf{EU}\,S_2$ is constrained too. Furthermore, from reduced expressions for S_1 and S_2, one can compute a reduced expression for S.*

By combining Lemmas 34, 35, 36, and 37, we obtain the result we were aiming at:

Theorem 38 *The model checking problem for $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}, \mathbf{EU}_\alpha)$ formulae is decidable for lossy channel systems.*

PROOF. First note that $\mathbf{EF}\,\varphi \equiv \mathtt{tt}\,\mathbf{EU}\,\varphi$ and $\mathbf{EF}_\tau\,\varphi \equiv \mathtt{tt}\,\mathbf{EU}_\tau\,\varphi$. For a given formula $\varphi \in \mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}, \mathbf{EU}_\alpha)$, the model checking algorithm computes a reduced expression which encodes the set of all configurations that satisfy φ. This is achieved by replacing each subformula of φ with its associated reduced expression in a bottom-up fashion, using algorithms of Lemmas 34, 35, 36, and 37. □

6 Applications

6.1 A Note on Semantic Quotients

Definition 39 *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 of the existing process equivalences we have that $s \sim [s]$ for every process s (see [19,20]). In general, the class of modal properties preserved under $\sim$-quotients is larger than the class of $\sim$-invariant properties [20]. 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 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?

Question (a) is known as *the strong regularity problem* (see, e.g., [18] where it is shown that strong regularity wrt. simulation equivalence is decidable for one-counter nets). For bisimulation-like equivalences, question (a) coincides with the standard regularity problem.

Using the results of previous sections, 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 PS equivalence. Further, let us assume that the states of $\mathcal{F}$ are pairwise non-equivalent (this can be effectively checked). Consider the formula

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

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

Observe that if problem (b) above is decidable for a given class of processes, then problem (a) is semi-decidable 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 semi-decidable due to rather generic reasons, while establishing the semi-decidability of the negative subcase is a model-specific part of the problem.

6.2 Results for Concrete Process Classes

All of the results presented so far 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 [39]. 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) [40];
- PA (and in fact also PAD) processes [27,25]. 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 nonprimitive recursive complexity [32].

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 [26]);
- 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 [32]).

Similar results hold for PS equivalences where P is well-defined (for pushdown processes we obtain **EXPSPACE** 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. *Information and Computation*, 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. *Information and Computation*, 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. *Journal of the Association for Computing Machinery*, 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 *Lecture Notes in Computer Science*, 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 *Lecture Notes in Computer Science*, 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 *Lecture Notes in Computer Science*, pages 323–333. Springer, 1999.

[7] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the Association for Computing Machinery*, 30(2):323–342, 1983.

[8] M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59(1–2):115–131, 1988.

[9] O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification on infinite structures. *Handbook of Process Algebra*, pages 545–623, 1999.

[10] R. de Nicola and F. Vaandrager. Three logics for branching bisimulation. *Journal of the Association for Computing Machinery*, 42(2):458–487, 1995.

[11] J. Esparza and M. Nielsen. Decidability issues for Petri nets — a survey. *Journal of Information Processing and Cybernetics*, 30(3):143–160, 1994.

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

[13] Y. Hirshfeld and M. Jerrum. Bisimulation equivalence is decidable for normed process algebra. In *Proceedings of ICALP'99*, volume 1644 of *Lecture Notes in Computer Science*, pages 412–421. Springer, 1999.

[14] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theoretical Computer Science*, 158(1–2):143–159, 1996.

[15] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimulation equivalence of normed basic parallel processes. *Mathematical Structures in Computer Science*, 6(3):251–259, 1996.

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

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

[18] P. Jančar, A. Kučera, and F. Moller. Simulation and bisimulation over one-counter processes. In *Proceedings of STACS'2000*, volume 1770 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 2000.

[19] A. Kučera. On finite representations of infinite-state behaviours. *Information Processing Letters*, 70(1):23–30, 1999.

[20] A. Kučera and J. Esparza. A logical viewpoint on process-algebraic quotients. *Journal of Logic and Computation*, 13(6):863–880, 2003.

[21] A. Kučera and P. Jančar. Equivalence-checking on infinite-state systems: Techniques and results. *Theory and Practice of Logic Programming*. To Appear.

[22] A. Kučera and R. Mayr. On the complexity of semantic equivalences for pushdown automata and BPA. In *Proceedings of MFCS 2002*, volume 2420 of *Lecture Notes in Computer Science*, pages 433–445. Springer, 2002.

[23] A. Kučera and R. Mayr. Simulation preorder over simple process algebras. *Information and Computation*, 173(2):184–198, 2002.

[24] 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*, pages 395–408. Kluwer, 2004.

[25] D. Lugiez and Ph. Schnoebelen. The regular viewpoint on PA-processes. *Theoretical Computer Science*, 274(1–2):89–115, 2002.

[26] R. Mayr. On the complexity of bisimulation problems for pushdown automata. In *Proceedings of IFIP TCS'2000*, volume 1872 of *Lecture Notes in Computer Science*, pages 474–488. Springer, 2000.

[27] R. Mayr. Decidability of model checking with the temporal logic EF. *Theoretical Computer Science*, 256(1–2):31–62, 2001.

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

[29] M. Müller-Olm. Derivation of characteristic formulae. *Electronic Notes in Theoretical Computer Science*, 18, 1998.

[30] J. Parrow and P. Sjödin. Multiway synchronization verified with coupled simulation. In *Proceedings of CONCUR'92*, volume 630 of *Lecture Notes in Computer Science*, pages 518–533. Springer, 1992.

[31] J. Parrow and P. Sjödin. The complete axiomatization of cs-congruence. In *Proceedings of STACS'94*, volume 775 of *Lecture Notes in Computer Science*, pages 557–568. Springer, 1994.

[32] Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

[33] G. Sénizergues. L(A)=L(B)? Decidability results from complete formal systems. *Theoretical Computer Science*, 251(1–2):1–166, 2001.

[34] J. Srba. Roadmap of infinite results. *EATCS Bulletin*, 78:163–175, 2002.

[35] B. Steffen and A. Ingólfsdóttir. Characteristic formulae for processes with divergence. *Information and Computation*, 110(1):149–163, 1994.

[36] 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 *Lecture Notes in Computer Science*, pages 66–81. Springer, 1993.

[37] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the Association for Computing Machinery*, 43(3):555–600, 1996.

[38] M. Voorhoeve and S. Mauw. Impossible futures and determinism. *Information Processing Letters*, 80(1):51–58, 2001.

[39] I. Walukiewicz. Model checking CTL properties of pushdown systems. In *Proceedings of FST&TCS'2000*, volume 1974 of *Lecture Notes in Computer Science*, pages 127–138. Springer, 2000.

[40] I. Walukiewicz. Private communication, September 2003.