FI MU

Faculty of Informatics
Masaryk University

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

by

Antonín Kučera
Philippe Schnoebelen

FI MU Report Series **FIMU-RS-2004-05**

 June 2004

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

Antonín Kučera*
Faculty of Informatics,
Masaryk University,
Botanická 68a, CZ-60200 Brno,
Czech Republic,
tony@fi.muni.cz.

Philippe Schnoebelen
LSV, ENS de Cachan & CNRS UMR 8643,
61, av. Pdt. Wilson,
94235 Cachan Cedex, France.
phs@lsv.ens-cachan.fr

Abstract

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

1 Introduction

Verification of infinite-state models of systems is a very active field of research, see [EN94, BCMS01, Bou01, KJ02, Srb02] for surveys of some subfields. In this area, researchers consider a large variety of models suited to

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

different kinds of applications, and three main kinds of verification problems: (1) specific properties like reachability or termination, (2) model checking of temporal formulae, and (3) semantic equivalences or preorders between two systems. With most models, termination and reachability are investigated first. Positive results lead to investigations of more general temporal model checking problems. Regarding equivalence problems, positive decidability results exist mainly for strong bisimilarity (some milestones in the study include [BBK93, HJM96b, HJM96a, Jan95, HJ99, Sén01]). For other behavioral equivalences, results are usually negative.

Regular equivalence problem. Recently, the problem of comparing some infinite-state process g with a *finite-state* specification f has been identified as an important subcase[1] of the general equivalence checking problem [KJ02]. 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 [KJ02].

The literature offers two generic techniques for deciding regular equivalences. First, Abdulla *et al.* show how to check *regular simulation* on *well-structured* processes [AČJT00]. Their algorithm is generic because a large collection of infinite-state models are well-structured [FS01].

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 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-

[1] 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.

like equivalences if one uses the modal μ-calculus [SI94, MO98]. Browne *et al.* constructed characteristic formulae wrt. bisimilarity and branching-bisimilarity in the logic CTL [BCG88]. 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 [JKM01]. This logic is sufficiently simple and its associated model-checking problem is decidable in many classes of infinite-state systems (including PA, lossy channel systems, and pushdown automata) [May01].

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

We offer two main reductions. One applies to a large parameterized family of equivalences defined via a transfer property (we call them MTB equivalences). The other applies to a large parameterized family of equivalences based on sets of enriched traces (we call them PQ equivalences). Together they cover virtually all process equivalences used in verification [vG93]. 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. More precisely, the constructions output a *characteristic formula* for f wrt. a given equivalence, which expresses the property

[2] 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.

of "being fully equivalent to f". In particular, this works for bisimulation-like equivalences (weak, delay, early, branching), and thus we also obtain a refinement of the result presented in [BCG88] which says that a characteristic formula wrt. branching bisimilarity is constructible in CTL. The main "message" of this part is that full regular equivalence is decidable for many more semantic equivalences and classes of infinite-state models than regular equivalence. In this paper we do not aim to develop specific methods for particular models and equivalences. (Such methods can be more efficient than our generic (model-independent) algorithm—for example, it has recently been shown in [KM04] that full regular equivalence with PDA processes can be decided by a PDA-specific algorithm which needs only polynomial time for some MTB equivalences and some subclasses of PDA processes.)

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

Plan of the paper. We introduce and discuss full regular equivalence in Section 2. In Section 3 we introduce MTB equivalences, show how to approximate them, and how to use these approximations to reformulate the condition of full regular equivalence into simpler but equivalent conditions (Theorem 3.6). In Section 3.1 we show how to encode the simplified conditions of Theorem 3.6 into modal logic. We also consider associated complexity questions. We introduce PQ equivalences in Section 4 and show a similar simplified way of checking the conditions of full regular equivalence. This is encoded into modal logic in Section 4.1. The model-checking algorithm for $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha)$ and lossy channel systems is presented in Section 5. This study brings a number of corollaries that are summarized at the end of 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 $\alpha, \beta, \ldots$ to range over Act_τ. 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 2.1 (Full Regular Equivalence). *We say* g *is* fully equivalent *to* f *(in* $\mathcal{F}$*) iff:*

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

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

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

encoded as regular bisimulation equivalence or regular simulation equivalence by any finite-state system. Let us also note that when $\sim$ is an equivalence of the bisimulation family, then regular equivalence is automatically "full".

3 MTB Preorder and Equivalence

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

A *transfer* is one of the three operators on binary relations defined as follows: $sim(R) = R$, $bisim(R) = R \cap R^{-1}$, $contrasim(R) = R^{-1}$. 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 3.1. *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 $s = s_0 \stackrel{\tau}{\to} \cdots \stackrel{\tau}{\to} s_k \stackrel{\alpha}{\to} t_0 \stackrel{\tau}{\to} \cdots \stackrel{\tau}{\to} t_\ell = t$, where $k, \ell \geq 0$, such that the following holds: (1) if $\eta \in M$, then $(s_i, s_j) \in \mathcal{S}$ for all $0 \leq i,j \leq k$; (2) if $d \in M$, then $(t_i, t_j) \in \mathcal{S}$ for all $0 \leq i,j \leq \ell$.*

The loose $\mathcal{S}$-consistency *of $s \stackrel{\alpha}{\Rightarrow} t$ 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 3.2. *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 base).

Lemma 3.3. *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 $k, \ell \geq 0$, 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 $(p', q') \in \mathsf{T}(\mathcal{R})$. 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 $(p', q') \in \mathsf{T}(\preccurlyeq)$ (hence also $(p', q') \in \mathsf{T}(\mathcal{R})$). 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. □

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

- $(s, t) \in$ *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).

The mode specifies the level of 'control' over the states that are passed through by $\stackrel{\alpha}{\Rightarrow}$ transitions. In particular, by putting $\mathsf{T} =$ *bisim*, $\mathsf{B} =$ *true*, and choosing M to be $\emptyset$, $\{\eta\}$, $\{d\}$, or $\{\eta, d\}$, one obtains weak bisimilarity [Mil89], η-bisimilarity [BvG87], delay-bisimilarity, and branching bisimilarity [vGW96], respectively.[3] "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 $\mathsf{T} =$ *sim*, $\mathsf{B} =$ *true*, and $\mathsf{M} = \emptyset$, we obtain weak simulation equivalence;

[3] 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.

and by redefining B to *ready* we yield a variant of ready simulation equivalence. The equivalence where $T = \mathit{contrasim}$, $B = \mathit{true}$, and $M = \emptyset$ is known as contrasimulation (see, e.g., [VM01]).[4]

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 3.4. *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 $s \stackrel{\alpha}{\Rightarrow} s'$ there is some tightly $\sqsubseteq_k$-consistent move $t \stackrel{\alpha}{\Rightarrow} t'$ 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 3.5. *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 $F \times F$.*

Proof. Since $\sqsubseteq_{k+1}$ refines $\sqsubseteq_k$, 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 3.3. □

Theorem 3.6. *Let $\mathcal{F} = (F, \rightarrow, \mathcal{A})$ be a finite-state system with n states, f a process of F, and g some (arbitrary) process. Then the following three conditions are equivalent.*

(a) $g \sim f$ *and for every* $g \rightarrow^* g'$ *there is some* $f' \in F$ *such that* $g' \sim f'$.

[4] Contrasimulation can also be seen as a generalization of coupled simulation [PS92, PS94], which was defined only for the subclass of divergence-free processes (where it coincides with contrasimulation). It is worth to note that contrasimulation coincides with strong bisimilarity on the subclass of 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.

(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 3.3). 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 $\bar{g} \stackrel{\alpha}{\Rightarrow} \bar{g}'$ be a tightly $\sqsubseteq_{n^2}$-consistent move. By definition of $\sqsubseteq_{n^2}$, there is a tightly $\sqsubseteq_{n^2-1}$-consistent move $\bar{f} \stackrel{\alpha}{\Rightarrow} \bar{f}'$ such that $(\bar{g}', \bar{f}') \in T(\sqsubseteq_{n^2-1})$. First, realize that the move $\bar{f} \stackrel{\alpha}{\Rightarrow} \bar{f}'$ is also tightly $\sqsubseteq_{n^2}$-consistent, because $\sqsubseteq_{n^2-1} = \sqsubseteq_{n^2}$ over $F \times F$ (see Lemma 3.5). 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 3.5. 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 $\bar{f} \stackrel{\alpha}{\Rightarrow} \bar{f}'$ 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 $\bar{g} \stackrel{\alpha}{\Rightarrow} \bar{g}'$ such that $(\bar{f}', \bar{g}') \in T(\sqsubseteq_{n^2-1})$. Now it suffices to show that

(1) the move $\bar{g} \stackrel{\alpha}{\Rightarrow} \bar{g}'$ 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 $\bar{g} \stackrel{\alpha}{\Rightarrow} \bar{g}'$ 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 3.5. 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 3.3. $\square$

3.1 Encoding MTB Equivalence into Modal Logic

In this section we show that the conditions (b) and (c) of Theorem 3.6 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):

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

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 $s' \models \varphi$.
- $s \models \varphi_1\,\mathbf{EU}_\alpha\,\varphi_2$ 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 [Mil89];
- $\mathcal{L}(\mathbf{EU}_\alpha)$ is the logic proposed by de Nicola and Vaandrager in [dNV95] which modally characterizes branching bisimilarity;
- $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ is the logic used in [JKM01] 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 3.6. Clearly such formulae cannot always exist without some additional assumptions about the base B. Actually, all we need is to assume that the equivalence B with processes of a given finite-state system $\mathcal{F} = (F, \rightarrow, \mathcal{A})$ is definable in the aforementioned logics. More precisely, for each $f \in F$ there should be formulae 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 $(g, f) \in B$ iff $g \models \Xi^t_f$ iff $g \models \Xi^\ell_f$. 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 3.7. *We say that a base* B *is* well-defined *if there is a polynomial* $\mathcal{P}$ *(in two variables) such that for every finite-state system* $\mathcal{F} = (F, \rightarrow, \mathcal{A})$ *the set* $\{\Xi^t_f, \Xi^\ell_f \mid f \in F\}$ *can be computed, and the relation* $B \cap (F \times F)$ *can be decided, in time* $\mathcal{O}(\mathcal{P}(|F|, |\mathcal{A}|))$.

Remark 3.8. *Note that a well-defined* B *is not necessarily decidable over process classes which contain infinite-state processes—for example, the ready base introduced in the previous section is well-defined but it is not decidable for, e.g., CCS processes. In fact, the* 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 base of the previous section, which is definable in* $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ *but not in* $\mathcal{L}(\mathbf{EF}, \mathbf{EU}_\alpha)$. □

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

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

- $s \models \langle \alpha, \varphi_\eta, \varphi_d \rangle^t \varphi$ iff either $\alpha = \tau$ and $s \models \varphi$, or there is a sequence of the form $s = p_0 \xrightarrow{\tau} \cdots p_k \xrightarrow{\alpha} q_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} q_m$, 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 3.9. *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

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

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

□

Since the conditions (b) and (c) of Theorem 3.6 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, $\mathrel{\mathring{=}}_k$ denotes either $\sim_k$ or $\approx_k$, and $\preccurlyeq_k$ 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 3.10. *For all $f \in F$ and $k \in \mathbb{N}_0$ we define the formulae $\Phi_{f,k}$, $\Psi_{f,k}$, and $\Theta_{f,k}$ inductively as follows:*

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

where

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

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

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 3.11. *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* $g' \mathrel{\dot{=}}_k f'$.

(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* $g' \mathrel{\dot{=}}_k f'$.

(c) $g \models \Theta_{f,0}$ *iff* $g \mathrel{\dot{=}}_0 f$; *further,* $g \models \Theta_{f,k+1}$ *iff* $f \mathrel{\dot{=}}_{k+1} g$ *and for each* $g \rightarrow^* g'$ *there is* $f' \in F$ *such that* $g' \mathrel{\dot{=}}_k f'$.

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 $g' \mathrel{\dot{=}}_k f'$, we can apply induction hypotheses

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

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

To see this, realize that for each $f \xrightarrow{\alpha,k} f'$ there is some $g \xrightarrow{\alpha,k} g'$ 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 $g \mathrel{\dot{=}}_k f_1$ and $g' \mathrel{\dot{=}}_k f_2$. As $g \xrightarrow{\alpha,k} g'$, we can apply 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 $f \xrightarrow{\alpha,k} f'$, hence $g \models \bigwedge_{f \xrightarrow{\alpha,k} f'} (\bigvee_{f_1,f_2 \in F} \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle \xi_{f',k})$ as needed.

For the '$\Rightarrow$' 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 induction hypothesis to conclude that for every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \mathrel{\dot{=}}_k f'$. It remains to show that $f \leqslant_{k+1} g$. Clearly $(f, g) \in B$ because $g \models \Xi_f$. Let $f \xrightarrow{\alpha,k} f'$. 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 induction hypothesis we obtain that there is $g \xrightarrow{\alpha,k} g'$ such that $g' \models \xi_{f',k}$, which means $(f', g') \in T(\leqslant_k)$.

(b) '$\Leftarrow$': Let us assume that $g \leqslant_{k+1} f$ and for each $g \rightarrow^* g'$ there is $f' \in F$ such that $g' \mathrel{\dot{=}}_k f'$. 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 $g \models [\alpha, \varphi_{f_1,k}, \psi_{f_2,k}](\bigvee_{f \xrightarrow{\alpha,k} f'} \rho_{f',k})$. Suppose the converse, i.e., $g \models \langle \alpha, \varphi_{f_1,k}, \psi_{f_2,k} \rangle (\bigwedge_{f \xrightarrow{\alpha,k} f'} \neg \rho_{f',k})$. By applying induction hypothesis we obtain that there is $g \xrightarrow{\alpha,k} g'$ such that for every $f \xrightarrow{\alpha,k} f'$ we have $g' \not\models \rho_{f',k}$, i.e., $(g', f') \notin T(\leqslant_k)$. Hence, $g \not\leqslant_{k+1} f$ which is a contradiction.

'$\Rightarrow$': 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 $g' \mathrel{\dot{=}}_k f'$ (by induction hypothesis). We show that $g \leqslant_{k+1} f$. Let $g \xrightarrow{\alpha,k} g'$. Since g, g' are reachable from g, there are $f_1, f_2 \in F$ such that $g \mathrel{\dot{=}}_k f_1$ and $g' \mathrel{\dot{=}}_k f_2$. Since $g \models [\alpha, \varphi_{f_1,k}, \psi_{f_2,k}](\bigvee_{f \xrightarrow{\alpha,k} f'} \rho_{f',k})$, we have that $g' \models \bigvee_{f \xrightarrow{\alpha,k} f'} \rho_{f',k}$ by using induction hypothesis. Hence, there is $f \xrightarrow{\alpha,k} f'$ such that $g' \models \rho_{f',k}$, 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 $\stackrel{\circ}{=}_k$ equivalence with g and g'. Since g and g' can be infinite-state processes, this is generally impossible. This difficulty was overcome in Theorem 3.11 by using the assumption that g and g' are $\stackrel{\circ}{=}_k$ equivalent to some f_1 and f_2 of F. Thus, we only needed to encode the $\stackrel{\circ}{=}_k$ 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 3.6 and Theorem 3.11 is the following:

Corollary 3.12. *Let g be an (arbitrary) process such that $\mathcal{A}(g) \subseteq \mathcal{A}$, and let $f \in \mathsf{F}$. Then the following two conditions are equivalent:*

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

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

Since the formula $\Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in \mathsf{F}} \Theta_{f',n^2})$ is effectively constructible, the problem (a) of the previous corollary is effectively reducible to the problem (b).

Remark 3.13. *An important consequence of Corollary 3.12 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[KM02b], 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 [May01, LS02]. Another example are pushdown processes. Model-checking $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau)$ for PDA is in **PSPACE** [Wal00]. 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 [KM02a]. 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 be possibly avoided.*

A natural question is what is the complexity of the reduction from (a) to (b). At first glance, it seems to be exponential because the size of 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 $\Theta_{f,n^2} \wedge \mathbf{AG}(\bigvee_{f' \in \mathsf{F}} \Theta_{f',n^2})$ by a *circuit*[5], then the size of this circuit is only

[5] 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.

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 3.15. We start by proving an auxiliary technical lemma:

Lemma 3.14. *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 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 3.15. *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.*

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 3.14. 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 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 3.12 and Lemma 3.15 can also be applied to finite-state processes (i.e., to processes of some finite-state system $\mathcal{F}$).

Corollary 3.16. *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 3.12 (b). If $M = \emptyset$, there is no need to construct the $\leqslant_k$ relations, because $\xrightarrow{\alpha,k} = \stackrel{\alpha}{\Rightarrow}$ for every $k \in \mathbb{N}_0$. Hence, the construction of the formula of Corollary 3.12 (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 PQ Preorder and Equivalence

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 $s \stackrel{\alpha}{\Rightarrow} t$. In the next definition we introduce another parametrized equivalence which is an abstract template for trace-like equivalences.

Definition 4.1. *Let* P *be a preorder over the class of all processes and let* $Q \in \{\forall, \exists\}$*. For every* $i \in \mathbb{N}_0$ *we inductively define the relation* $\sqsubseteq_i$ *as follows:*

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

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

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

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

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

Lemma 4.2. *Let* $\mathcal{F} = (F, \rightarrow, \mathcal{A})$ *be a finite-state system with* n *states. Then* $\sqsubseteq_{n2^n-1} = \sqsubseteq_{n2^n} = \sqsubseteq$*, where all of the relations are considered as being restricted to* $F \times 2^F$*.*

Lemma 4.3. *For all* $i \in \mathbb{N}_0$*, processes* s, t*, and sets of processes* M, N *we have that*

(a) if $s \sqsubseteq_i t$ *and* $t \sqsubseteq_i M$*, then also* $s \sqsubseteq_i M$*;*

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

Theorem 4.4. *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 two 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) $g \sim_{n2^n} f$ *and for every* $g \to^* g'$ *there is some* $f' \in F$ *such that* $g' \sim_{n2^n} f'$.

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

$g' \not\sqsubseteq_m \bar{f}$**.** By definition of $\sqsubseteq_i$ (and the fact that $m > n2^n$), there must be some $g' \to^* g''$ and $M \subseteq F$ such that $g'' \sqsubseteq_{n2^n-1} M$ and $g'' \not\sqsubseteq_{n2^n} M$. We show that this is impossible. To see this, realize that $g \to^* g''$ and due to (b) there is some $f' \in F$ such that $g'' \sim_{n2^n} f'$. So, $f' \sqsubseteq_{n2^n} g'' \sqsubseteq_{n2^n-1} M$, which means $f' \sqsubseteq_{n2^n-1} M$ by Lemma 4.3 (a). Hence, $f' \sqsubseteq_{n2^n} M$ by Lemma 4.2. Now $g'' \sqsubseteq_{n2^n} f' \sqsubseteq_{2^n} M$ and thus we obtain $g'' \sqsubseteq_{n2^n} M$ by applying Lemma 4.3 (a), which is a contradiction.

$\bar{f} \not\sqsubseteq_m g'$**.** Then there must be some $\bar{f} \to^* f'$ and a set of processes M such that every $g'' \in M$ is reachable from g', $f' \sqsubseteq_{n2^n-1} M$, and $f' \not\sqsubseteq_{n2^n} M$. 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 $g'' \sim_{n2^n} f''$, and vice versa. Hence, $f' \sqsubseteq_{n2^n-1} N$ by Lemma 4.3 (b), which means that $f' \sqsubseteq_{n2^n} N$ by Lemma 4.2. Thus, we obtain $f' \sqsubseteq_{n2^n} M$ again by applying Lemma 4.3 (b) (the roles of M, N are interchanged now), which is a contradiction. □

Now we show how to encode the condition (b) of Theorem 4.4 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 4.5. *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 S preorder is (provably) not well-defined. Nevertheless, our results *do* apply to possible-futures equivalence, as we shall see in Remark 4.10.

Lemma 4.6. *If* P *is well-defined, then the relation* $\sqsubseteq_i$ *over* $F \times 2^F$ *can be computed in time which is exponential in* n *and polynomial in* i*.*

4.1 Encoding PQ Preorder into Modal Logic

Definition 4.7. *For all* $i \in \mathbb{N}_0$*,* $f \in F$*, and* $M \subseteq F$ *we define the sets*

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

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

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

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 4.6), hence the $\Phi_{f,k}$, ..., formulae are effectively constructible too.

Theorem 4.8. *Let g be an (arbitrary) process such that $\mathcal{A}(g) \subseteq \mathcal{A}$. Then for all $f \in F$ and $k \in \mathbb{N}_0$ we have the following:*

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

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

(c) $g \models \Theta_{f,0}$ *iff* $g \stackrel{\circ}{=}_0 f$*; further,* $g \models \Theta_{f,k+1}$ *iff* $f \sim_{k+1} g$ *and for each* $g \rightarrow^* g'$ *there is* $f' \in F$ *such that* $g' \sim_k f'$.

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), $\Longrightarrow$" Let $g \models \Phi_{f,k+1}$. Then $g \models \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ and hence for every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \sim_k f'$ by applying induction hypothesis. We show that $f \sqsubseteq_{k+1} g$. As $g \models \Xi_f$, we have that $(f, g) \in P$. Let $f \stackrel{\alpha}{\Rightarrow} f'$. Since $g \models \bigwedge_{f \stackrel{\alpha}{\Rightarrow} f'} (\bigvee_{M \in \mathcal{F}(f', \sqsubseteq_k)} (\bigwedge_{f'' \in M} \langle\!\langle \alpha \rangle\!\rangle \Theta_{f'',k}))$, there is $M \subseteq F$ such that $f' \sqsubseteq_k M$ (this follows from the definition of $\mathcal{F}(f', \sqsubseteq_k)$). 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 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 $g_i \sim_k f_i$. Note that $g \stackrel{\alpha}{\Rightarrow} N$. We claim that $f' \sqsubseteq_k N$. However, this follows immediately from Lemma 4.3 (b).

  "(a), $\Longleftarrow$" Let us assume that $f \sqsubseteq_{k+1} g$ and for every $g \rightarrow^* g'$ there is $f' \in F$ such that $g' \sim_k f'$. Then $g \models \Xi_f \wedge \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ by applying the definition of $\sqsubseteq_{k+1}$ and induction hypothesis. Since $f \sqsubseteq_{k+1} g$, for every $f \stackrel{\alpha}{\Rightarrow} f'$ there is some $g \stackrel{\alpha}{\Rightarrow} N$ such that $f' \sqsubseteq_k N$. Now let $M = \{f'' \in F \mid f'' \sim_k g'' \text{ for some } g'' \in N\}$. Since every state of N is reachable from g, for every $g'' \in N$ there is at least one $f'' \in M$ such that $g'' \sim_k f''$. As $f' \sqsubseteq_k N$, we also have that $f' \sqsubseteq_k M$ by applying Lemma 4.3 (b). Hence, $M \in \mathcal{F}(f', \sqsubseteq_k)$. To sum up, we obtain that $g \models \bigwedge_{f \stackrel{\alpha}{\Rightarrow} f'} (\bigvee_{M \in \mathcal{F}(f', \sqsubseteq_k)} (\bigwedge_{f'' \in M} \langle\!\langle \alpha \rangle\!\rangle \Theta_{f'',k}))$ and we are done.

  "(b), $\Longrightarrow$" Let $g \models \Psi_{f,k+1}$. Then $g \models \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ and hence for

every $g \rightarrow^* g'$ there is some $f' \in F$ such that $g' \sim_k f'$ by applying induction hypothesis. We show that $g \sqsubseteq_{k+1} f$. As $g \models \Gamma_f$, we have that $(g, f) \in P$. Let $g \stackrel{\alpha}{\Rightarrow} g'$. Since $g \models \bigwedge_{\alpha \in \mathcal{A}_\tau} [\![\alpha]\!] (\bigvee_{f \stackrel{\alpha}{\Rightarrow} M} \bigvee_{f' \in \mathcal{F}(\sqsubseteq_k, M)} \Theta_{f',k})$, there are $f \stackrel{\alpha}{\Rightarrow} M$ and $f' \in F$ such that $f' \sqsubseteq_k M$ and $g' \sim_k f'$ (here we apply the definition of $\mathcal{F}(\sqsubseteq_k, M)$ and induction hypothesis). Since $g' \sqsubseteq_k f' \sqsubseteq_k M$, we obtain $g' \sqsubseteq_k M$ by Lemma 4.3 (a).

"(b), $\Longleftarrow$" Let us assume that $g \sqsubseteq_{k+1} f$ and for every $g \rightarrow^* g'$ there is $f' \in F$ such that $g' \sim_k f'$. Then $g \models \Gamma_f \wedge \mathbf{AG} \bigvee_{f' \in F} \Theta_{f',k}$ by applying the definition of $\sqsubseteq_{k+1}$ and induction hypothesis. Since $g \sqsubseteq_{k+1} f$, for every $g \stackrel{\alpha}{\Rightarrow} g'$ there is some $f \stackrel{\alpha}{\Rightarrow} M$ such that $g' \sqsubseteq_k M$. Further, as g' is reachable from g, there is some $f' \in F$ such that $g' \sim_k f'$. Since $f' \sqsubseteq_k g' \sqsubseteq_k M$, we obtain $f' \sqsubseteq_k M$ by Lemma 4.3 (a). This means that $f' \in \mathcal{F}(\sqsubseteq_k, M)$. To sum up, we have that $g \models \bigwedge_{\alpha \in \mathcal{A}_\tau} [\![\alpha]\!] (\bigvee_{f \stackrel{\alpha}{\Rightarrow} M} \bigvee_{f' \in \mathcal{F}(\sqsubseteq_k, M)} \Theta_{f',k})$ and the proof is finished. □

Corollary 4.9. *Let g be an (arbitrary) process such that $\mathcal{A}(g) \subseteq \mathcal{A}$, and let $f \in F$. Then the following two conditions are equivalent:*

(a) $g \sim f$ *and for every* $g \rightarrow^* g'$ *there is some* $f' \in F$ *such that* $g' \sim f'$.

(b) $g \models \Theta_{f,n2^n} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n2^n})$.

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

Remark 4.10. *As we already mentioned, the S preorder is not well-defined, because trace equivalence with a given finite-state process f is not expressible in modal logic (even monadic second order logic is (provably) not sufficiently powerful to express that a process can perform every trace over a given finite alphabet). Nevertheless, in our context it suffices to express the condition of* full trace equivalence *with f, which is achievable. So, full possible-futures equivalence with f is expressed by the formula $\Theta_{f,n2^n} \wedge \mathbf{AG}(\bigvee_{f' \in F} \Theta_{f',n2^n})$ where for every $f' \in F$ we define $\Xi_{f'}$ and $\Gamma_{f'}$ to be the formula which expresses full trace equivalence with f'. This "trick" can be used also for other trace-like equivalences where the associated P is not well-defined.*

5 Model checking lossy channel systems

In this section we show that the model checking of $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha)$ formulae is decidable for lossy channel systems (LCS's). This result was inspired by [BM99] and can be seen as a natural extension of known results.

We refer to [AJ96, Sch02] for motivations and definitions on LCS's. Here we only need to know that a *configuration* σ of a LCS C is a pair $\langle q, w\rangle$ of a control state q from some finite set Q and a finite word $w \in \Sigma^*$ describing the current contents of the channel (for simplicity we assume a single channel). Here $\Sigma = \{a, b, \ldots\}$ is a finite alphabet of messages. The behavior of C is given by a transition system $\mathcal{T}_C$ where steps $\sigma \to \sigma'$ describe how the configuration can evolve. In the rest of this section, we assume a fixed LCS C.

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

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

Lemma 5.1 (see [AJ96, Sch02]). *If $\sigma \to \sigma'$ is a step of $\mathcal{T}_C$, then for all configurations $\theta \geq \sigma$ and $\theta' \leq \sigma'$, $\theta \to \theta'$ is a step of $\mathcal{T}_C$ too.*

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 $\uparrow\sigma - \uparrow\theta_1 - \cdots - \uparrow\theta_n$ (for some configurations $\theta_1, \ldots, \theta_n$). This denotes an upward-closure minus some restrictions (the $\uparrow\theta_i$'s).

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 $\rho_1 \vee \cdots \vee \rho_m$. Such an expression is *reduced* if no ρ_i is trivial. For a set S of configurations, $\mathit{Pre}(S) = \{\sigma \mid \exists \theta \in S, \sigma \to \theta\}$ is the set of (immediate) predecessors of configurations in S.

We now show that constrained sets are closed under Boolean operations, and that there exist effective algorithms reducing expressions like

$\gamma_1 \wedge \gamma_2$ or $\neg\gamma$ to an equivalent reduced expression. Additionally, constrained sets are effectively closed under *Pre*. This makes them a suitable representation for symbolic model checking.

Lemma 5.2. *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. The intersection $\uparrow\langle q_1, w_1\rangle \wedge \uparrow\langle q_2, w_2\rangle$ of two upward-closures is empty when $q_1 \neq q_2$. Otherwise it is computed by a simple enumeration. 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 $\uparrow\sigma \wedge \uparrow\sigma' = \uparrow\sigma_1 \vee \ldots \vee \uparrow\sigma_l$, one derives

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

This allows intersecting constrained sets: $(\bigvee_i \rho_i) \wedge (\bigvee_j \rho_j) = \bigvee_i \bigvee_j (\rho_i \wedge \rho_j)$. □

Lemma 5.3. *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:

$$\neg\uparrow\langle q, w\rangle = (\uparrow\langle q, \epsilon\rangle - \uparrow\langle q, w\rangle) \vee \bigvee_{q' \neq q} \uparrow\langle q', \epsilon\rangle.$$

This allows complementing restricted sets:

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

We use intersection (Lemma 5.2) for complementing constrained sets:

$$\neg(\rho_1 \vee \cdots \vee \rho_m) = (\neg\rho_1) \wedge \cdots \wedge (\neg\rho_m).$$

□

Lemma 5.4. *Constrained sets are closed under immediate predecessors. Furthermore, from a reduced expression γ, one can compute a reduced expression for Pre(γ).*

Sketch. Since $Pre(\bigvee_i \rho_i) = \bigvee_i Pre(\rho_i)$, it is enough to compute $Pre(\rho)$ for ρ a restricted set. Now, if ρ has the reduced form $\uparrow\sigma - \uparrow\theta_1 - \ldots - \uparrow\theta_n$, then $Pre(\rho) = Pre(\uparrow\sigma)$ (by Fact 5.1). With the methods of [AJ96], one easily produces a union of upward-closures for this set. □

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

Lemma 5.5. *Let S_1 and S_2 be two constrained sets. Then the set S of configurations that satisfy S_1 **EU** S_2 is constrained too. Furthermore, from reduced expressions for S_1 and S_2, one can compute a reduced expression for S.*

Proof. We inductively define a sequence $(U_i)_{i\in\mathbb{N}_0}$ of sets of configurations with $U_0 = S_2$ and $U_{i+1} = U_i \cup (S_1 \cap Pre(U_i))$. Then $S = \bigcup_i U_i$.

By the previous Lemmas, every U_i is a constrained set and one can compute, for each $S_1 \cap Pre(U_i)$, a reduced expression $\bigvee_j \rho_{i,j}$ with $\rho_{i,j}$ having the form $\uparrow\sigma_{i,j} - \uparrow\theta_{i,j,1} - \ldots - \uparrow\theta_{i,j,k}$. 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* (Lemma 5.4) does not use restrictions, and the algorithm for intersection (see, Eq. (1) in Lemma5.2) 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 s.t. ρ_{i,j_i} is not included in U_i. Extract from the sequence $(\rho_{i,j_i})_i$ 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 $\rho_{i',j_{i'}}$, 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. □

By combining Lemmas 5.3, 5.4 and 5.5, we obtain the result we were aiming at:

Corollary 5.6. *Let φ be a modal formula in $\mathcal{L}(\mathbf{EX}, \mathbf{EU})$. The set of configurations that satisfy φ is a constrained set, and one can compute a reduced expression for this set.*

Theorem 5.7. *The model checking problem for $\mathcal{L}(\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha)$ formulae is decidable for lossy channel systems.*

Proof. When C has labeled rules, it is easy to deal with modalities from $\{\mathbf{EX}_\alpha, \mathbf{EF}, \mathbf{EF}_\tau, \mathbf{EU}_\alpha\}$. One simply disregards rules carrying a wrong label for reducing $\mathbf{EX}_\alpha$ to $\mathbf{EX}$, or $\mathbf{EF}_\tau$ to $\mathbf{EF}$. For $\mathbf{EU}_a$ we reduce $\varphi_1 \mathbf{EU}_a \varphi_2$ to $\varphi_1 \mathbf{EU}(\varphi_1 \wedge \mathbf{EX}_a \varphi_2)$. □

6 Applications

A Note on Semantic Quotients. Let $\mathcal{T} = (S, \rightarrow, \mathcal{A})$ be a transition system, $g \in S$, and $\sim$ a process equivalence. Let $\mathit{Reach}(g) = \{s \in S \mid g \rightarrow^* s\}$. The $\sim$-quotient of g is the process $[g]$ of the transition system $(\mathit{Reach}(g)/\sim, \rightarrow, \mathcal{A})$ where $[s] \xrightarrow{\alpha} [t]$ iff there are $s', t' \in \mathit{Reach}(g)$ such that $s \sim s'$, $t \sim t'$, and $s' \xrightarrow{\alpha} t'$.

For most (if not all) of the existing process equivalences we have that $s \sim [s]$ for every process s (see [Kuč99, KE03]). In general, the class of temporal properties preserved under $\sim$-quotients is larger than the class of $\sim$-invariant properties [KE03]. Hence, $\sim$-quotients are rather robust descriptions of the original systems. Some questions related to formal verification can be answered by examining the properties of $\sim$-quotients, which is particularly advantageous if the $\sim$-quotient is finite (so far, mainly the bisimilarity-quotients have been used for this purpose). This raises two natural problems:

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

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

The question (a) is known as *the strong regularity problem* (see, e.g., [JKM00] where it is shown that strong regularity wrt. simulation equivalence is decidable for one-counter nets). For bisimulation-like equivalences, the question (a) coincides with the standard regularity problem.

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

effectively checked). Consider the formula

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

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

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

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

- pushdown processes. In fact, this problem is **PSPACE**-complete [Wal00]. 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) [Wal03];
- PA (and in fact also PAD) processes [May01, LS02]. The best known complexity upper bound for this problem in non-elementary.
- lossy channel systems (see Section 5). Here the model-checking problem is of nonprimitive recursive complexity.

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

- decidable in polynomial space for pushdown processes. For many concrete $\mathcal{MTB}$-equivalences, this bound is optimal (for example, all bisimulation-like equivalences between pushdown processes and finite-state processes are **PSPACE**-hard [May00]);
- 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 [Sch02]).

Similar results hold for PQ-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

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

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

[BBK93] 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.

[BCG88] 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.

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

[BM99] 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.

[Bou01] 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.

[BvG87] 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.

[dNV95] R. de Nicola and F. Vaandrager. Three logics for branching bisimulation. *Journal of the Association for Computing Machinery*, 42(2):458–487, 1995.

[EN94] J. Esparza and M. Nielsen. Decidability issues for Petri nets — a survey. *Journal of Information Processing and Cybernetics*, 30(3):143–160, 1994.

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

[HJ99] 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.

[HJM96a] 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.

[HJM96b] 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.

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

[JKM00] 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.

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

[KE03] A. Kučera and J. Esparza. A logical viewpoint on process-algebraic quotients. *Journal of Logic and Computation*, 13(6):863–880, 2003.

[KJ02] A. Kučera and P. Jančar. Equivalence-checking with infinite-state systems: Techniques and results. In *Proceedings of SOFSEM'2002*, volume 2540 of *Lecture Notes in Computer Science*, pages 41–73. Springer, 2002.

[KM02a] 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.

[KM02b] A. Kučera and R. Mayr. Simulation preorder over simple process algebras. *Information and Computation*, 173(2):184–198, 2002.

[KM04] A. Kučera and R. Mayr. A generic framework for checking semantic equivalences between pushdown automata and finite-state automata. In *Proceedings of IFIP TCS'2004*. Kluwer, 2004. To appear.

[Kuč99] A. Kučera. On finite representations of infinite-state behaviours. *Information Processing Letters*, 70(1):23–30, 1999.

[LS02] D. Lugiez and Ph. Schnoebelen. The regular viewpoint on PA-processes. *Theoretical Computer Science*, 274(1–2):89–115, 2002.

[May00] 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.

[May01] R. Mayr. Decidability of model checking with the temporal logic EF. *Theoretical Computer Science*, 256(1–2):31–62, 2001.

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

[MO98] M. Müller-Olm. Derivation of characteristic formulae. *Electronic Notes in Theoretical Computer Science*, 18, 1998.

[PS92] 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.

[PS94] 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.

[Sch02] Ph. Schnoebelen. Verifying lossy channel systems has non-primitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

[Sén01] G. Sénizergues. L(A)=L(B)? Decidability results from complete formal systems. *Theoretical Computer Science*, 251(1–2):1–166, 2001.

[SI94] B. Steffen and A. Ingólfsdóttir. Characteristic formulae for processes with divergence. *Information and Computation*, 110(1):149–163, 1994.

[Srb02] J. Srba. Roadmap of infinite results. *EATCS Bulletin*, 78:163–175, 2002.

[vG93] 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.

[vGW96] 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.

[VM01] M. Voorhoeve and S. Mauw. Impossible futures and determinism. *Information Processing Letters*, 80(1):51–58, 2001.

[Wal00] 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.

[Wal03] I. Walukiewicz. Private communication, September 2003.

Publications in the FI MU Report Series are in general accessible via WWW and anonymous FTP:

`http://www.fi.muni.cz/informatics/reports/`
`ftp ftp.fi.muni.cz (cd pub/reports)`

Copies may be also obtained by contacting:

**Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic**