

Controller/Orchestrator Synthesis via Filtration

Philippe Balbiani

*CNRS — Université de Toulouse
Institut de recherche en informatique de Toulouse
118 ROUTE DE NARBONNE, 31062 TOULOUSE CEDEX 9, FRANCE
Philippe.Balbani@irit.fr*

Fahima Cheikh

*UT1 — Université de Toulouse
Institut de recherche en informatique de Toulouse
118 ROUTE DE NARBONNE, 31062 TOULOUSE CEDEX 9, FRANCE
Fahima.Cheikh@irit.fr*

Guillaume Feuillade

*UPS — Université de Toulouse
Institut de recherche en informatique de Toulouse
118 ROUTE DE NARBONNE, 31062 TOULOUSE CEDEX 9, FRANCE
Guillaume.Feuillade@irit.fr*

Abstract

The present paper is interested in the following decision problems: (1) given finite frames $\mathcal{F}, \mathcal{F}'$, determine if there exists a frame \mathcal{F}'' such that \mathcal{F} and $\mathcal{F}' \otimes \mathcal{F}''$, the synchronous product of \mathcal{F}' and \mathcal{F}'' , are bisimilar; (2) given finite frames $\mathcal{F}, \mathcal{F}'$, determine if there exists a frame \mathcal{F}'' such that \mathcal{F} and $\mathcal{F}' \oplus \mathcal{F}''$, the asynchronous product of \mathcal{F}' and \mathcal{F}'' , are bisimilar. It shows that variants of the filtration method are adequate for solving them.

Keywords: Modal logic, filtration, bisimulation, controller synthesis, orchestrator synthesis.

1 Introduction

Multifarious controller synthesis problems, as introduced by Maler *et al.* [11] and Ramadge and Wonham [14], amount, given finite transition systems $\mathcal{S}, \mathcal{S}'$, to determine if there exists a transition system \mathcal{S}'' such that \mathcal{S} and $\mathcal{S}' \otimes \mathcal{S}''$, the synchronous product of \mathcal{S}' and \mathcal{S}'' , are equivalent. The role of \mathcal{S}'' is to restrict the behaviours of \mathcal{S}' . Hence, in this setting, $\mathcal{S}, \mathcal{S}'$ and \mathcal{S}'' can be respectively seen as the control objective, the reactive system to be controlled and the controller whereas $\mathcal{S}' \otimes \mathcal{S}''$

denotes the restricted system. Controller synthesis problems arise in a variety of contexts ranging from computer operating systems to complex multimode processes. The exponential-time method proposed by Arnold *et al.* [3] to solve them consists in transforming them into formula satisfiability problems in μ -calculus [2].

Several orchestrator synthesis problems, as introduced by Berardi *et al.* [4] and Berardi *et al.* [5], amount, given finite distributed systems $\mathcal{S}, \mathcal{S}'$, to determine if there exists a distributed system \mathcal{S}'' such that \mathcal{S} and $\mathcal{S}' \oplus \mathcal{S}''$, the asynchronous product of \mathcal{S}' and \mathcal{S}'' , are equivalent. The role of \mathcal{S}'' is to enhance the behaviours of \mathcal{S}' . Hence, in this setting, $\mathcal{S}, \mathcal{S}'$ and \mathcal{S}'' can be respectively seen as the orchestration objective, the reactive system to be orchestrated and the orchestrator whereas $\mathcal{S}' \oplus \mathcal{S}''$ denotes the enhanced system. Orchestrator synthesis problems arise in a variety of contexts ranging from service oriented computing to ambient intelligence. The exponential-time method proposed by Berardi *et al.* [6] to solve them consists in transforming them into formula satisfiability problems in propositional dynamic logic [10].

Transition systems and distributed systems can be abstracted as frames. Hence, the present paper is interested in the following controller/orchestrator synthesis problems: (1) given finite frames $\mathcal{F}, \mathcal{F}'$, determine if there exists a frame \mathcal{F}'' such that \mathcal{F} and $\mathcal{F}' \otimes \mathcal{F}''$, the synchronous product of \mathcal{F}' and \mathcal{F}'' , are bisimilar; (2) given finite frames $\mathcal{F}, \mathcal{F}'$, determine if there exists a frame \mathcal{F}'' such that \mathcal{F} and $\mathcal{F}' \oplus \mathcal{F}''$, the asynchronous product of \mathcal{F}' and \mathcal{F}'' , are bisimilar. It is probably correct to say that these decision problems are motivated more by model-theoretic and complexity-theoretic characteristics than by tools for the philosophical analysis of modal concepts. Nevertheless, there are various reasons to believe that they are very similar to the formula satisfiability problems traditionally considered in modal logic.

What the present paper shows is that variants of the filtration method are adequate for solving them, i.e. we will use these variants to give exponential-time algorithms for solving our controller/orchestrator synthesis problems. Its section-by-section breakdown is as follows. Section 2 establishes the concepts of frame, bisimulation, synchronous product and asynchronous product. In section 3, basic definitions concerning the controller synthesis problem and the orchestrator synthesis problem are given. Based on variants of the filtration method, ways of solving both problems are presented in sections 4 and 5. Section 6 studies variants of our synthesis problems. We assume the reader is at home with tools and techniques in modal logic. For more on these see [7].

2 Basic notions

This section presents the basic notions needed to introduce the controller synthesis problem and the orchestrator synthesis problem.

2.1 Frame

Let PG be a set of program variables (with typical members denoted a, b , etc). A frame over PG is a structure of the form $\mathcal{F} = (W, R)$ where W is a nonempty set

of states (with typical members denoted x, y , etc) and R is a function from W^2 to 2^{PG} . The set W of states is to be regarded as the set of all possible states in a computational process whereas the function R from W^2 to 2^{PG} associates with each pair of states a set of program variables with $a \in R(x, y)$ meaning that state y can be reached from state x by performing program a . In this paper, we shall always consider that there exists a root $x_0 \in W$ such that for all $x \in W$, there exists a nonnegative integer n and $a_1, \dots, a_n \in PG$ such that x can be reached from x_0 by performing programs a_1, \dots, a_n . For all $a \in PG$, let $R_a \subseteq W \times W$ be the binary relation such that for all $x, y \in W$,

- $x R_a y$ iff $a \in R(x, y)$.

\mathcal{F} is said to be finite iff W is finite. We shall say that \mathcal{F} is deterministic iff for all $x \in W$, for all $a \in PG$, the set of all $y \in W$ such that $x R_a y$ has cardinality 0 or 1 whereas we shall say that \mathcal{F} is serial iff for all $x \in W$, for all $a \in PG$, the set of all $y \in W$ such that $x R_a y$ has cardinality 1 or more. \mathcal{F} is said to be an equivalence frame iff for all $a \in PG$, R_a is reflexive, symmetrical and transitive. We shall say that \mathcal{F} is reflexive (respectively: symmetrical, transitive) iff for all $a \in PG$, R_a is reflexive (respectively: symmetrical, transitive).

2.2 Bisimulation

Let $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ be frames over PG . A binary relation $Z \subseteq W \times W'$ is called a bisimulation between \mathcal{F} and \mathcal{F}' , in symbols $Z: \mathcal{F} \longleftrightarrow \mathcal{F}'$, iff for all $x \in W$, for all $x' \in W'$, if $x Z x'$ then

- for all $a \in PG$, for all $y \in W$, if $x R_a y$ then there exists $y' \in W'$ such that $x' R'_a y'$ and $y Z y'$,
- for all $a \in PG$, for all $y' \in W'$, if $x' R'_a y'$ then there exists $y \in W$ such that $x R_a y$ and $y Z y'$.

If $x \in W$, $x' \in W'$ are such that $x Z x'$ then we say that x and x' are bisimilar, in symbols $Z: \mathcal{F}, x \longleftrightarrow \mathcal{F}', x'$. If $x \in W$, $x' \in W'$ are such that there exists a bisimulation Z between \mathcal{F} and \mathcal{F}' such that $Z: \mathcal{F}, x \longleftrightarrow \mathcal{F}', x'$ then we write $\mathcal{F}, x \longleftrightarrow \mathcal{F}', x'$. It is a well-known fact that the following decision problem is in *PTIME*:

- Given a finite set PG of program variables, finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , roots $x_0 \in W$, $x'_0 \in W'$, determine whether $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}', x'_0$.

See [1] for details.

2.3 Synchronous product and asynchronous product

Let $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ be frames over PG . By $\mathcal{F} \otimes \mathcal{F}'$, we denote the synchronous product of \mathcal{F} and \mathcal{F}' , i.e. the frame $\mathcal{F}'' = (W'', R'')$ over PG where

- $W'' = W \times W'$,
- R'' is the function from W''^2 to 2^{PG} such that for all $(x, x'), (y, y') \in W''$,

$R''((x, x'), (y, y'))$ is the set of all $a \in PG$ such that $a \in R(x, y)$ and $a \in R'(x', y')$.

Let $\mathcal{G} = (V, E)$, $\mathcal{G}' = (V', E')$ be frames over PG , $x_0 \in W$, $x'_0 \in W'$, $v_0 \in V$, $v'_0 \in V'$ be roots. The proof of the following lemma is left to the reader.

Lemma 2.1 *If $\mathcal{F}, x_0 \longleftrightarrow \mathcal{G}, v_0$ and $\mathcal{F}', x'_0 \longleftrightarrow \mathcal{G}', v'_0$ then $\mathcal{F} \otimes \mathcal{F}', (x_0, x'_0) \longleftrightarrow \mathcal{G} \otimes \mathcal{G}', (v_0, v'_0)$.*

By $\mathcal{F} \oplus \mathcal{F}'$, we denote the asynchronous product of \mathcal{F} and \mathcal{F}' , i.e. the frame $\mathcal{F}'' = (W'', R'')$ over PG where

- $W'' = W \times W'$,
- R'' is the function from W''^2 to 2^{PG} such that for all $(x, x'), (y, y') \in W''$, $R''((x, x'), (y, y'))$ is the set of all $a \in PG$ such that $a \in R(x, y)$ and $x' = y'$ or $x = y$ and $a \in R'(x', y')$.

Let $\mathcal{G} = (V, E)$, $\mathcal{G}' = (V', E')$ be frames over PG , $x_0 \in W$, $x'_0 \in W'$, $v_0 \in V$, $v'_0 \in V'$ be roots. The proof of the following lemma is left to the reader.

Lemma 2.2 *If $\mathcal{F}, x_0 \longleftrightarrow \mathcal{G}, v_0$ and $\mathcal{F}', x'_0 \longleftrightarrow \mathcal{G}', v'_0$ then $\mathcal{F} \oplus \mathcal{F}', (x_0, x'_0) \longleftrightarrow \mathcal{G} \oplus \mathcal{G}', (v_0, v'_0)$.*

3 Controller synthesis and orchestrator synthesis

This section presents our controller/orchestrator synthesis problems. Let us consider a finite set PG of program variables.

3.1 Decision problems

Let $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ be finite frames over PG , $x_0 \in W$, $x'_0 \in W'$ be roots. Given a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$, we say that (\mathcal{F}'', x''_0) controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$. The synthesis of controllers is the following decision problem:

(SC) Given a finite set PG of program variables, finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , roots $x_0 \in W$, $x'_0 \in W'$, determine whether there exists a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$ such that (\mathcal{F}'', x''_0) controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) .

Given a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$, we say that (\mathcal{F}'', x''_0) orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}'', (x'_0, x''_0)$. The synthesis of orchestrators is the following decision problem:

(SO) Given a finite set PG of program variables, finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , roots $x_0 \in W$, $x'_0 \in W'$, determine whether there exists a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$ such that (\mathcal{F}'', x''_0) orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) .

(SC) and (SO) are deeply related to several important topics considered in the theory of controller synthesis [3,11,14] and in the theory of orchestrator synthesis [4,5,6]. In the theory of controller synthesis, the basic problem is to restrict, by

means of a controller, the behaviours of a given transition system, the reactive system to be controlled, so that it satisfies the given control objective. In the theory of orchestrator synthesis, the basic problem is to enhance, by means of an orchestrator, the behaviours of a given distributed system, the multiagent system to be orchestrated, so that it satisfies the given orchestration objective. In [3] and [6], methods consisting in transforming every instance of the controller synthesis problem or the orchestrator synthesis problem into an instance of the formula satisfiability problem in μ -calculus or the formula satisfiability problem in propositional dynamic logic are proposed. What sections 4 and 5 show is that alternative methods based on variants of the filtration method are adequate for solving (SC) and (SO).

3.2 Bisimulations and products

Let $\mathcal{F}_1'' = (W_1'', R_1'')$, $\mathcal{F}_2'' = (W_2'', R_2'')$ be frames over PG , $x_1'' \in W_1''$, $x_2'' \in W_2''$ be roots. The following lemma demonstrates that control and orchestration are invariant under bisimulations.

Lemma 3.1 *If $\mathcal{F}_1'', x_1'' \longleftrightarrow \mathcal{F}_2'', x_2''$ then for all finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , for all roots $x_0 \in W$, $x'_0 \in W'$,*

- *(\mathcal{F}_1'', x_1'') controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff (\mathcal{F}_2'', x_2'') controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0),*
- *(\mathcal{F}_1'', x_1'') orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff (\mathcal{F}_2'', x_2'') orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0).*

Proof. By lemmas 2.1 and 2.2. □

We say that

- \mathcal{F}_1'', x_1'' and \mathcal{F}_2'', x_2'' are control-equivalent, in symbols $\mathcal{F}_1'', x_1'' \equiv_c \mathcal{F}_2'', x_2''$, iff for all finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , for all roots $x_0 \in W$, $x'_0 \in W'$, (\mathcal{F}_1'', x_1'') controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff (\mathcal{F}_2'', x_2'') controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0),
- \mathcal{F}_1'', x_1'' and \mathcal{F}_2'', x_2'' are orchestration-equivalent, in symbols $\mathcal{F}_1'', x_1'' \equiv_o \mathcal{F}_2'', x_2''$, iff for all finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , for all roots $x_0 \in W$, $x'_0 \in W'$, (\mathcal{F}_1'', x_1'') orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff (\mathcal{F}_2'', x_2'') orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0).

The Hennessy-Milner theorem [7] states that modally equivalent image-finite models are bisimilar. The following lemmas show that control-equivalent frames are bisimilar and orchestration-equivalent frames are bisimilar.

Lemma 3.2 *If $\mathcal{F}_1'', x_1'' \equiv_c \mathcal{F}_2'', x_2''$ then $\mathcal{F}_1'', x_1'' \longleftrightarrow \mathcal{F}_2'', x_2''$.*

Proof. Suppose that $\mathcal{F}' = (W', R')$ is a finite frame over PG , $x'_0 \in W'$ is a root such that

- R' is the function from W'^2 to 2^{PG} such that
 - for all $a \in PG$, for all $x', y' \in W'$, $x' R'_a y'$.

The reader may easily verify that $\mathcal{F}_1'', x_1'' \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}_1'', (x'_0, x''_1)$ and $\mathcal{F}_2'', x_2'' \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}_2'', (x'_0, x''_2)$. Hence, $\mathcal{F}_1'', x_1'' \longleftrightarrow \mathcal{F}_2'', x_2''$. \square

Lemma 3.3 *If $\mathcal{F}_1'', x_1'' \equiv_o \mathcal{F}_2'', x_2''$ then $\mathcal{F}_1'', x_1'' \longleftrightarrow \mathcal{F}_2'', x_2''$.*

Proof. Suppose that $\mathcal{F}' = (W', R')$ is a finite frame over PG , $x'_0 \in W'$ is a root such that

- R' is the function from W'^2 to 2^{PG} such that
 - for all $a \in PG$, for all $x', y' \in W'$, not $x' R'_a y'$.

The reader may easily verify that $\mathcal{F}_1'', x_1'' \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}_1'', (x'_0, x''_1)$ and $\mathcal{F}_2'', x_2'' \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}_2'', (x'_0, x''_2)$. Hence, $\mathcal{F}_1'', x_1'' \longleftrightarrow \mathcal{F}_2'', x_2''$. \square

3.3 Deterministic/serial frames

Suppose that we are given finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , roots $x_0 \in W$, $x'_0 \in W'$. The following lemma shows that determining whether there exists a controller of (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) becomes easier if \mathcal{F} is deterministic or \mathcal{F}' is deterministic.

Lemma 3.4 *If \mathcal{F} is deterministic or \mathcal{F}' is deterministic then there exists a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$ such that (\mathcal{F}'', x''_0) controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff (\mathcal{F}, x_0) controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) .*

Proof. Let $\mathcal{F}'' = (W'', R'')$ be a frame over PG , $x''_0 \in W''$ be a root such that (\mathcal{F}'', x''_0) controls (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) . Hence, there exists a bisimulation Z between \mathcal{F} and $\mathcal{F}' \times \mathcal{F}''$ such that $Z: \mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$. Let $Z^s \subseteq W \times (W' \times W)$ be the binary relation such that for all $x_1 \in W$, for all $(x', x_2) \in W' \times W$, $x_1 Z^s (x', x_2)$ iff there exists $x \in W$, there exists $x'' \in W''$ such that $x_1 = x$, $x_2 = x$ and $x Z (x', x'')$. We demonstrate that $Z^s: \mathcal{F} \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}$. Let $x_1 \in W$, $(x', x_2) \in W' \times W$ be such that $x_1 Z^s (x', x_2)$.

Let $a \in PG$, $y \in W$ be such that $x_1 R_a y$. Since $x_1 Z^s (x', x_2)$, then there exists $x \in W$, there exists $x'' \in W''$ such that $x_1 = x$, $x_2 = x$ and $x Z (x', x'')$. Since $x_1 R_a y$, then $x R_a y$. Since $x_2 = x$, then $x_2 R_a y$. Since $x R_a y$ and $x Z (x', x'')$, then there exists $(y', y'') \in W' \times W''$ such that $x' R'_a y'$, $x'' R''_a y''$ and $y Z (y', y'')$. Hence, $y Z^s (y', y)$.

Let $a \in PG$, $(y', y) \in W' \times W$ be such that $x' R'_a y'$ and $x_2 R_a y$. Since $x_1 Z^s (x', x_2)$, then there exists $x \in W$, there exists $x'' \in W''$ such that $x_1 = x$, $x_2 = x$ and $x Z (x', x'')$. Since $x_2 R_a y$, then $x R_a y$. Since $x_1 = x$, then $x_1 R_a y$. Since $x Z (x', x'')$, then there exists $(z', z'') \in W' \times W''$ such that $x' R'_a z'$, $x'' R''_a z''$ and $y Z (z', z'')$. Since $x' R'_a y'$ and $x Z (x', x'')$, then there exists $z \in W$ such that $x R_a z$ and $z Z (y', z'')$. If \mathcal{F} is deterministic then $y = z$. Since $z Z (y', z'')$, then $y Z^s (y', y)$. If \mathcal{F}' is deterministic then $y' = z'$. Since $y Z (z', z'')$, then $y Z^s (y', y)$. \square

As a result,

Proposition 3.5 *If one considers instances $(PG, \mathcal{F}, \mathcal{F}', x_0, x'_0)$ of (SC) such that \mathcal{F} is deterministic or \mathcal{F}' is deterministic then (SC) is in PTIME.*

The following lemma shows that determining whether there exists an orchestrator of (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) becomes easier if \mathcal{F} is serial or \mathcal{F}' is serial.

Lemma 3.6 *If \mathcal{F} is serial or \mathcal{F}' is serial then there exists a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$ such that (\mathcal{F}'', x''_0) orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) iff (\mathcal{F}, x_0) orchestrates (\mathcal{F}', x'_0) within (\mathcal{F}, x_0) .*

Proof. The proof is similar to the proof of lemma 3.4. □

As a result,

Proposition 3.7 *If one considers instances $(PG, \mathcal{F}, \mathcal{F}', x_0, x'_0)$ of (SO) such that \mathcal{F} is serial or \mathcal{F}' is serial then (SO) is in $PTIME$.*

4 Deciding (SC)

In this section, we show that (SC) is in $EXPTIME$. We demonstrate the existence of an $EXPTIME$ algorithm using filtration.

4.1 Synchronous filtration

We now establish a simple algorithm for solving (SC) . This simple algorithm is based on a variant of the filtration method [7]. Suppose that we are given a finite set PG of program variables, finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , roots $x_0 \in W$, $x'_0 \in W'$. Let $\mathcal{F}'' = (W'', R'')$ be a frame over PG , $x''_0 \in W''$ be a root such that $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$. Hence, there exists a bisimulation Z between \mathcal{F} and $\mathcal{F}' \otimes \mathcal{F}''$ such that $x_0 Z (x'_0, x''_0)$. Let $\equiv \subseteq W'' \times W''$ be the binary relation such that for all $x''_1, x''_2 \in W''$,

- $x''_1 \equiv x''_2$ iff for all $x \in W$, for all $x' \in W'$, $x Z (x', x''_1)$ iff $x Z (x', x''_2)$.

Note that \equiv is an equivalence relation. Let $x'' \in W''$. The set of all states in W'' equivalent to x'' modulo \equiv , in symbols $|x''|$, is called the equivalence class of x'' in W'' modulo \equiv with x'' as its representative. The set of all equivalence classes of W'' modulo \equiv , in symbols W'' / \equiv , is called the quotient set of W'' modulo \equiv . Suppose that $\mathcal{F}^f = (W^f, R^f)$ is a frame over PG such that

- $W^f = W'' / \equiv$,
- R^f is a function from W^{f^2} to 2^{PG} such that
 - for all $a \in PG$, for all $x'', y'' \in W''$, if there exists $z'', t'' \in W''$ such that $x'' \equiv z'', y'' \equiv t''$ and $z'' R''_a t''$ then $|x''| R^f_a |y''|$,
 - for all $a \in PG$, for all $x'', y'' \in W''$, if $|x''| R^f_a |y''|$ then for all $x \in W$, for all $x', y' \in W'$, if $x' R'_a y'$ and $x Z (x', x'')$ then there exists $y \in W$ such that $x R_a y$ and $y Z (y', y'')$.

Then \mathcal{F}^f is called a filtration of \mathcal{F}'' through \mathcal{F} and \mathcal{F}' . Remark that $Card(W^f) \leq 2^{Card(W) \times Card(W')}$. Let $Z^f \subseteq W \times (W' \times W^f)$ be the binary relation such that for all $x \in W$, for all $(x', |x''|) \in W' \times W^f$, $x Z^f (x', |x''|)$ iff $x Z (x', x'')$. It is a simple matter to check that

Lemma 4.1 $Z^f: \mathcal{F} \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^f$.

Proof. Let $x \in W$, $(x', |x''|) \in W' \times W^f$ be such that $x Z^f (x', |x''|)$. Hence, $x Z (x', x'')$.

Let $a \in PG$, $y \in W$ be such that $x R_a y$. We demonstrate that there exists $(y', |y''|) \in W' \times W^f$ such that $x' R'_a y'$, $|x''| R^f_a |y''|$ and $y Z^f (y', |y''|)$. Since $x Z (x', x'')$, then there exists $(y', y'') \in W' \times W''$ such that $x' R'_a y'$, $x'' R''_a y''$ and $y Z (y', y'')$. Hence, there exists $(y', |y''|) \in W' \times W^f$ such that $x' R'_a y'$, $|x''| R^f_a |y''|$ and $y Z^f (y', |y''|)$.

Let $a \in PG$, $(y', |y''|) \in W' \times W^f$ be such that $x' R'_a y'$ and $|x''| R^f_a |y''|$. We demonstrate that there exists $y \in W$ such that $x R_a y$ and $y Z^f (y', |y''|)$. Since $x Z (x', x'')$, then there exists $y \in W$ such that $x R_a y$ and $y Z (y', y'')$. Hence, there exists $y \in W$ such that $x R_a y$ and $y Z^f (y', |y''|)$. \square

Hence,

Lemma 4.2 $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^f, (x'_0, |x''_0|)$.

There are at least two ways to define functions R^f from W^{f^2} to 2^{PG} that fulfil the required conditions. Define the functions R^f_{inf} and R^f_{sup} from W^{f^2} to 2^{PG} as follows:

- for all $a \in PG$, for all $|x''|, |y''| \in W^f$, $a \in R^f_{inf}(|x''|, |y''|)$ iff there exists $z'', t'' \in W''$ such that $x'' \equiv z''$, $y'' \equiv t''$ and $a \in R''(z'', t'')$,
- for all $a \in PG$, for all $|x''|, |y''| \in W^f$, $a \in R^f_{sup}(|x''|, |y''|)$ iff for all $x \in W$, for all $x', y' \in W'$, if $a \in R'(x', y')$ and $x Z (x', x'')$ then there exists $y \in W$ such that $a \in R(x, y)$ and $y Z (y', y'')$.

Lemma 4.3 R^f_{inf} and R^f_{sup} satisfy the two conditions of a filtration.

Proof. By definition, R^f_{inf} satisfies the first condition of a filtration.

Let $a \in PG$, $x'', y'' \in W''$ be such that $|x''| R^f_{inf_a} |y''|$. We demonstrate that for all $x \in W$, for all $x', y' \in W'$, if $x' R'_a y'$ and $x Z (x', x'')$ then there exists $y \in W$ such that $x R_a y$ and $y Z (y', y'')$. Let $x \in W$, $x', y' \in W'$ be such that $x' R'_a y'$ and $x Z (x', x'')$. Since $|x''| R^f_{inf_a} |y''|$, then there exists $z'', t'' \in W''$ such that $x'' \equiv z''$, $y'' \equiv t''$ and $a \in R''(z'', t'')$. Since $x Z (x', x'')$, then $x Z (x', z'')$. Since $x' R'_a y'$ and $a \in R''(z'', t'')$, then there exists $y \in W$ such that $x R_a y$ and $y Z (y', t'')$. Since $y'' \equiv t''$, then $y Z (y', y'')$.

Let $a \in PG$, $x'', y'' \in W''$ be such that there exists $z'', t'' \in W''$ such that $x'' \equiv z''$, $y'' \equiv t''$ and $z'' R''_a t''$. We demonstrate that $|x''| R^f_{sup_a} |y''|$. Let $x \in W$, $x', y' \in W'$ be such that $a \in R'(x', y')$ and $x Z (x', x'')$. Since $x'' \equiv z''$, then $x Z (x', z'')$. Since $a \in R'(x', y')$ and $z'' R''_a t''$, then there exists $y \in W$ such that $a \in R(x, y)$ and $y Z (y', t'')$. Since $y'' \equiv t''$, then $y Z (y', y'')$.

By definition, R^f_{sup} satisfies the second condition of a filtration. \square

Lemma 4.4 For all $|x''|, |y''| \in W^f$, $R^f_{inf}(|x''|, |y''|) \subseteq R^f_{sup}(|x''|, |y''|)$.

Proof. Let $a \in PG$ be such that $a \in R_{inf}^f(|x''|, |y''|)$. We demonstrate that $a \in R_{sup}^f(|x''|, |y''|)$. Let $x \in W$, $x', y' \in W'$ be such that $a \in R'(x', y')$ and $x Z (x', x'')$. We demonstrate that there exists $y \in W$ such that $a \in R(x, y)$ and $y Z (y', y'')$. Since $a \in R_{inf}^f(|x''|, |y''|)$, then there exists $z'', t'' \in W''$ such that $x'' \equiv z''$, $y'' \equiv t''$ and $a \in R''(z'', t'')$. Since $x Z (x', x'')$, then $x Z (x', z'')$. Since $a \in R'(x', y')$ and $a \in R''(z'', t'')$, then there exists $y \in W$ such that $a \in R(x, y)$ and $y Z (y', t'')$. Since $y'' \equiv t''$, then there exists $y \in W$ such that $a \in R(x, y)$ and $y Z (y', y'')$. \square

From the discussion above, it follows that the functions R_{inf}^f and R_{sup}^f from W^{f2} to 2^{PG} give respectively rise to the least filtration $\mathcal{F}_{inf}^f = (W^f, R_{inf}^f)$ of \mathcal{F}'' through \mathcal{F} and \mathcal{F}' and the greatest filtration $\mathcal{F}_{sup}^f = (W^f, R_{sup}^f)$ of \mathcal{F}'' through \mathcal{F} and \mathcal{F}' .

4.2 Complexity of (SC)

In this section, we show how the synchronous filtration can be used for deciding (SC).

4.2.1 A nondeterministic exponential-time algorithm

For our purpose, the crucial property of the above notion of synchronous filtration is the following: $Card(W^f) \leq 2^{Card(W) \times Card(W')}$. Hence, we can give a simple algorithm for solving (SC): guess a frame $\mathcal{F}'' = (W'', R'')$ over PG such that $Card(W'') \leq 2^{Card(W) \times Card(W')}$, guess a root $x''_0 \in W''$ and determine whether $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$. Not surprisingly, the above algorithm returns the value *true* iff there exists a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$ such that $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$. Seeing that determining whether $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$ can be done in polynomial time [1], it follows immediately that

Proposition 4.5 (SC) is in NEXPTIME.

4.2.2 A deterministic exponential-time algorithm

The truth of the matter is that (SC) is in EXPTIME. This can be proved as follows. Suppose that we are given a finite set PG of program variables, finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , roots $x_0 \in W$, $x'_0 \in W'$. Let $\mathcal{F}'' = (W'', R'')$ be a frame over PG , $x''_0 \in W''$ be a root such that $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$. Hence, there exists a bisimulation Z between \mathcal{F} and $\mathcal{F}' \otimes \mathcal{F}''$ such that $x_0 Z (x'_0, x''_0)$. Let f be the function from W'' to $2^{W \times W'}$ such that for all $x'' \in W''$,

- $f(x'') = \{(x, x') \in W \times W' : x Z (x', x'')\}$.

By definition, for all $x''_1, x''_2 \in W''$, if $f(x''_1) = f(x''_2)$ then $x''_1 \equiv x''_2$. Suppose that $\mathcal{F}^{ff} = (W^{ff}, R^{ff})$ is the frame over PG such that

- $W^{ff} = f(W'')$,
- R^{ff} is the function from W^{ff2} to 2^{PG} such that
 - for all $a \in PG$, for all $x'', y'' \in W''$, $f(x'') R^{ff}_a f(y'')$ iff for all $x \in W$, for all

$x', y' \in W'$, if $x' R'_a y'$ and $x Z (x', x'')$ then there exists $y \in W$ such that $x R_a y$ and $y Z (y', y'')$.

Let $Z^{ff} \subseteq W \times (W' \times W^{ff})$ be the binary relation such that for all $x \in W$, for all $(x', f(x'')) \in W' \times W^{ff}$, $x Z^{ff} (x', f(x''))$ iff $x Z^f (x', |x''|)$. It is a simple matter to check that

Lemma 4.6 $Z^{ff}: \mathcal{F} \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^{ff}$.

Proof. By lemmas 4.1 and 4.3. □

Hence,

Lemma 4.7 $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^{ff}, (x'_0, f(x''_0))$.

We now construct a sequence $\mathcal{F}^i = (W^i, R^i)$, $i \geq 0$, of frames over PG approximating $\mathcal{F}^{ff} = (W^{ff}, R^{ff})$ and a sequence $Z^i \subseteq W \times (W' \times W^i)$, $i \geq 0$, of binary relations approximating Z^{ff} .

Let $\mathcal{F}^0 = (W^0, R^0)$ be the frame over PG such that

- $W^0 = 2^{W \times W'}$,
- R^0 is the function from W^{02} to 2^{PG} such that
 - for all $a \in PG$, for all $x^0, y^0 \in W^0$, $x^0 R^0_a y^0$ iff for all $x \in W$, for all $x', y' \in W'$, if $x' R'_a y'$ and $(x, x') \in x^0$ then there exists $y \in W$ such that $x R_a y$ and $(y, y') \in y^0$,

$Z^0 \subseteq W \times (W' \times W^0)$ be the binary relation such that for all $x \in W$, for all $(x', x^0) \in W' \times W^0$, $x Z^0 (x', x^0)$ iff $(x, x') \in x^0$.

Secondly, for all $i \geq 0$, let $W^i_{\rightarrow} = \{x^i \in W^i: \text{there exists } a \in PG, x \in W, x' \in W', y \in W \text{ such that } x Z^i (x', x^i), x R_a y \text{ and for all } y' \in W', \text{ for all } y^i \in W^i, \text{ if } x' R'_a y' \text{ and } x^i R^i_a y^i \text{ then not } y Z^i (y', y^i)\}$, $W^i_{\leftarrow} = \{x^i \in W^i: \text{there exists } a \in PG, x \in W, x' \in W', y' \in W', y^i \in W^i \text{ such that } x Z^i (x', x^i), x' R'_a y', x^i R^i_a y^i \text{ and for all } y \in W, \text{ if } x R_a y \text{ then not } y Z^i (y', y^i)\}$, $\mathcal{F}^{i+1} = (W^{i+1}, R^{i+1})$ be the frame over PG such that

- $W^{i+1} = W^i \setminus (W^i_{\rightarrow} \cup W^i_{\leftarrow})$,
- R^{i+1} is the function from W^{i+12} to 2^{PG} such that
 - for all $a \in PG$, for all $x^{i+1}, y^{i+1} \in W^{i+1}$, $x^{i+1} R^{i+1}_a y^{i+1}$ iff for all $x \in W$, for all $x', y' \in W'$, if $x' R'_a y'$ and $(x, x') \in x^{i+1}$ then there exists $y \in W$ such that $x R_a y$ and $(y, y') \in y^{i+1}$,

$Z^{i+1} \subseteq W \times (W' \times W^{i+1})$ be the binary relation such that for all $x \in W$, for all $(x', x^{i+1}) \in W' \times W^{i+1}$, $x Z^{i+1} (x', x^{i+1})$ iff $(x, x') \in x^{i+1}$.

Lemma 4.8 For all $i \geq 0$,

- $W^{ff} \subseteq W^i$,
- for all $x^{ff}, y^{ff} \in W^{ff}$, $R^{ff}(x^{ff}, y^{ff}) \subseteq R^i(x^{ff}, y^{ff})$,
- $Z^{ff} \subseteq Z^i$.

Proof. The proof is by induction on $i \geq 0$. As the reader is asked to show, $W^{ff} \subseteq W^0$, for all $x^{ff}, y^{ff} \in W^{ff}$, $R^{ff}(x^{ff}, y^{ff}) \subseteq R^0(x^{ff}, y^{ff})$ and $Z^{ff} \subseteq Z^0$. Let $i \geq 0$ be such that $W^{ff} \subseteq W^i$, for all $x^{ff}, y^{ff} \in W^{ff}$, $R^{ff}(x^{ff}, y^{ff}) \subseteq R^i(x^{ff}, y^{ff})$ and $Z^{ff} \subseteq Z^i$. We demonstrate that $W^{ff} \subseteq W^{i+1}$, for all $x^{ff}, y^{ff} \in W^{ff}$, $R^{ff}(x^{ff}, y^{ff}) \subseteq R^{i+1}(x^{ff}, y^{ff})$ and $Z^{ff} \subseteq Z^{i+1}$.

Let $x^{ff} \in W^{ff}$. If $x^{ff} \notin W^{i+1}$ then $x^{ff} \in W_{\rightarrow}^i$ or $x^{ff} \in W_{\leftarrow}^i$. If $x^{ff} \in W_{\rightarrow}^i$ then there exists $a \in PG$, $x \in W$, $x' \in W'$, $y \in W$ such that $x Z^i (x', x^{ff})$, $x R_a y$ and for all $y' \in W'$, for all $y^i \in W^i$, if $x' R'_a y'$ and $x^{ff} R_a^i y^i$ then not $y Z^i (y', y^i)$. Since $x Z^i (x', x^{ff})$, then $(x, x') \in x^{ff}$. Hence, $x Z^{ff} (x', x^{ff})$. Since $x R_a y$, then there exists $y' \in W'$, $y^i \in W^i$ such that $x' R'_a y'$ and $x^{ff} R_a^i y^i$ and $y Z^{ff} (y', y^i)$. Hence, $(y, y') \in y^i$. Hence, $y Z^i (y', y^i)$: a contradiction. If $x^{ff} \in W_{\leftarrow}^i$ then there exists $a \in PG$, $x \in W$, $x' \in W'$, $y' \in W'$, $y^i \in W^i$ such that $x Z^i (x', x^{ff})$, $x' R'_a y'$, $x^{ff} R_a^i y^i$ and for all $y \in W$, if $x R_a y$ then not $y Z^i (y', y^i)$. Since $x Z^i (x', x^{ff})$, then $(x, x') \in x^{ff}$. Since $x' R'_a y'$ and $x^{ff} R_a^i y^i$, then there exists $y \in W$ such that $x R_a y$ and $(y, y') \in y^i$. Hence, $y Z^i (y', y^i)$: a contradiction.

Let $a \in PG$ be such that $x^{ff} R^{ff}_a y^{ff}$. We demonstrate that $x^{ff} R^{i+1}_a y^{ff}$. Let $x \in W$, $x', y' \in W'$, be such that $x' R'_a y'$ and $(x, x') \in x^{ff}$. Hence, $x Z^{ff} (x', x^{ff})$. Since $x' R'_a y'$ and $x^{ff} R^{ff}_a y^{ff}$, then there exists $y \in W$ such that $x R_a y$ and $y Z^{ff} (y', y^{ff})$. Hence, $(y, y') \in y^{ff}$. Hence, $x^{ff} R^{i+1}_a y^{ff}$.

Let $x \in W$, $(x', x^{ff}) \in W' \times W^{ff}$ be such that $x Z^{ff} (x', x^{ff})$. We demonstrate that $x Z^{i+1} (x', x^{ff})$. Since $x Z^{ff} (x', x^{ff})$, then $(x, x') \in x^{ff}$. Hence, $x Z^{i+1} (x', x^{ff})$. \square

It follows that there exists $i_0 \geq 0$ such that

Lemma 4.9 $Z^{i_0}: \mathcal{F} \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^{i_0}$.

Proof. Since W^0 is finite and for all $i \geq 0$, $W^{i+1} \subseteq W^i$, then there exists $i_0 \geq 0$ such that $W^{i_0+1} = W^{i_0}$. Since for all $i \geq 0$, $W^{ff} \subseteq W^i$, then W^{i_0} is nonempty. Hence, $\mathcal{F}^{i_0} = (W^{i_0}, R^{i_0})$ is a frame over PG . Since $W^{i_0+1} = W^{i_0}$, then $W_{\rightarrow}^{i_0} = \emptyset$ and $W_{\leftarrow}^{i_0} = \emptyset$. Hence, $Z^{i_0}: \mathcal{F} \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^{i_0}$. \square

Hence,

Lemma 4.10 $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^{i_0}, (x'_0, f(x''_0))$.

The above construction has the following property. When applied to an arbitrary finite set PG of program variables, arbitrary finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , arbitrary roots $x_0 \in W$, $x'_0 \in W'$, it stops with a frame $\mathcal{F}^{i_0} = (W^{i_0}, R^{i_0})$ over PG and a binary relation $Z^{i_0} \subseteq W \times (W' \times W^{i_0})$ such that if $W^{i_0} \neq \emptyset$ then $Z^{i_0}: \mathcal{F} \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^{i_0}$. Hence, we can give a simple algorithm for solving (SC):

- For all $i \geq 0$, construct the frame $\mathcal{F}^i = (W^i, R^i)$ over PG and the binary relation $Z^i \subseteq W \times (W' \times W^i)$ as above until if $W^i \neq \emptyset$ then $Z^i: \mathcal{F} \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}^i$.

Not surprisingly, the above algorithm returns the value *true* iff there exists a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$ such that $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \otimes \mathcal{F}'', (x'_0, x''_0)$. Seeing that \mathcal{F}^0 and Z^0 can be constructed in exponential time and for all $i \geq 0$,

\mathcal{F}^{i+1} and Z^{i+1} can be constructed in time polynomial in the size of \mathcal{F}^i and Z^i , it follows immediately that

Proposition 4.11 *(SC) is in EXPTIME.*

5 Deciding (SO)

In this section, we show that (SO) is in EXPTIME. We demonstrate the existence of an EXPTIME algorithm using filtration.

5.1 Asynchronous filtration

We now establish a simple algorithm for solving (SO). This simple algorithm is based on a variant of the filtration method [7] similar to the one used in section 4.1. Suppose that we are given a finite set PG of program variables, finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , roots $x_0 \in W$, $x'_0 \in W'$. Let $\mathcal{F}'' = (W'', R'')$ be a frame over PG , $x''_0 \in W''$ be a root such that $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}'', (x'_0, x''_0)$. Hence, there exists a bisimulation Z between \mathcal{F} and $\mathcal{F}' \oplus \mathcal{F}''$ such that $x_0 Z (x'_0, x''_0)$. Defining $\mathcal{F}^f = (W^f, R^f)$ and $Z^f \subseteq W \times (W' \times W'')$ as in section 4.1 aside from the fact that the conditions put on the function R^f from W^{f^2} to 2^{PG} are now the following:

- for all $a \in PG$, for all $x'', y'' \in W''$, if there exists $z'', t'' \in W''$ such that $x'' \equiv z'', y'' \equiv t''$ and $a \in R''(z'', t'')$ then $a \in R^f(|x''|, |y''|)$,
- for all $a \in PG$, for all $x'', y'' \in W''$, if $a \in R^f(|x''|, |y''|)$ then for all $x \in W$, for all $x', y' \in W'$, if $x' = y'$ and $x Z (x', x'')$ then there exists $y \in W$ such that $a \in R(x, y)$ and $y Z (y', y'')$,

it is a simple matter to check that

Lemma 5.1 $Z^f: \mathcal{F} \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^f$.

Proof. The proof is similar to the proof of lemma 4.1. □

Hence,

Lemma 5.2 $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^f, (x'_0, |x''_0|)$.

As in section 4.1, there are at least two ways to define functions R^f from W^{f^2} to 2^{PG} that fulfil the required conditions:

- for all $a \in PG$, for all $|x''|, |y''| \in W^f$, $a \in R_{inf}^f(|x''|, |y''|)$ iff there exists $z'', t'' \in W''$ such that $x'' \equiv z'', y'' \equiv t''$ and $a \in R''(z'', t'')$,
- for all $a \in PG$, for all $|x''|, |y''| \in W^f$, $a \in R_{sup}^f(|x''|, |y''|)$ iff for all $x \in W$, for all $x', y' \in W'$, if $x' = y'$ and $x Z (x', x'')$ then there exists $y \in W$ such that $a \in R(x, y)$ and $y Z (y', y'')$.

Lemma 5.3 R_{inf}^f and R_{sup}^f satisfy the two conditions of a filtration.

Proof. The proof is similar to the proof of lemma 4.3. □

Lemma 5.4 For all $|x''|, |y''| \in W^f$, $R_{inf}^f(|x''|, |y''|) \subseteq R_{sup}^f(|x''|, |y''|)$.

Proof. The proof is similar to the proof of lemma 4.4. \square

From the discussion above, it follows that the functions R_{inf}^f and R_{sup}^f from W^{f^2} to 2^{PG} give respectively rise to the least filtration $\mathcal{F}_{inf}^f = (W^f, R_{inf}^f)$ of \mathcal{F}'' through \mathcal{F} and \mathcal{F}' and the greatest filtration $\mathcal{F}_{sup}^f = (W^f, R_{sup}^f)$ of \mathcal{F}'' through \mathcal{F} and \mathcal{F}' .

5.2 Complexity of (SO)

In this section, we show how the asynchronous filtration can be used for deciding (SO).

5.2.1 A nondeterministic exponential-time algorithm

For our purpose, the crucial property of the above notion of synchronous filtration is the following: $Card(W^f) \leq 2^{Card(W) \times Card(W')}$. Hence, we can give a simple nondeterministic exponential-time algorithm for solving (SO) similar to the one considered in section 4.2.1.

Proposition 5.5 (SO) is in NEXPTIME.

5.2.2 A deterministic exponential-time algorithm

The truth of the matter is that (SO) is in EXPTIME. This can be proved in a way similar to the one followed in section 4.2.2. Defining $\mathcal{F}^{ff} = (W^{ff}, R^{ff})$ and $Z^{ff} \subseteq W \times (W' \times W^{ff})$ as in section 4.2 aside from the fact that the definition of the function R^{ff} from W^{ff^2} to 2^{PG} is now the following:

- for all $a \in PG$, for all $x'', y'' \in W''$, $f(x'') R_a^{ff} f(y'')$ iff for all $x \in W$, for all $x', y' \in W'$, if $x' = y'$ and $x Z (x', x'')$ then there exists $y \in W$ such that $x R_a y$ and $y Z (y', y'')$,

it is a simple matter to check that

Lemma 5.6 $Z^{ff}: \mathcal{F} \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^{ff}$.

Proof. The proof is similar to the proof of lemma 4.6. \square

Hence,

Lemma 5.7 $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^{ff}, (x'_0, f(x''_0))$.

We now construct a sequence $\mathcal{F}^i = (W^i, R^i)$, $i \geq 0$, of frames over PG approximating $\mathcal{F}^{ff} = (W^{ff}, R^{ff})$ and a sequence $Z^i \subseteq W \times (W' \times W^i)$, $i \geq 0$, of binary relations approximating Z^{ff} .

Let $\mathcal{F}^0 = (W^0, R^0)$ be the frame over PG such that

- $W^0 = 2^{W \times W'}$,
- R^0 is the function from W^{0^2} to 2^{PG} such that

- for all $a \in PG$, for all $x^0, y^0 \in W^0$, $x^0 R_a^0 y^0$ iff for all $x \in W$, for all $x', y' \in W'$, if $x' = y'$ and $(x, x') \in x^0$ then there exists $y \in W$ such that $x R_a y$ and $(y, y') \in y^0$,

$Z^0 \subseteq W \times (W' \times W^0)$ be the binary relation such that for all $x \in W$, for all $(x', x^0) \in W' \times W^0$, $x Z^0 (x', x^0)$ iff $(x, x') \in x^0$.

Secondly, for all $i \geq 0$, let $W_{\rightarrow}^i = \{x^i \in W^i: \text{there exists } a \in PG, x \in W, x' \in W', y \in W \text{ such that } x Z^i (x', x^i), x R_a y \text{ and for all } y' \in W', \text{ for all } y^i \in W^i, \text{ if } x' R_a' y' \text{ and } x^i = y^i \text{ or } x' = y' \text{ and } x^i R_a' y^i \text{ then not } y Z^i (y', y^i)\}$, $W_{\leftarrow}^i = \{x^i \in W^i: \text{there exists } a \in PG, x \in W, x' \in W', y' \in W', y^i \in W^i \text{ such that } x Z^i (x', x^i), x' R_a' y' \text{ and } x^i = y^i \text{ or } x' = y' \text{ and } x^i R_a' y^i \text{ and for all } y \in W, \text{ if } x R_a y \text{ then not } y Z^i (y', y^i)\}$, $\mathcal{F}^{i+1} = (W^{i+1}, R^{i+1})$ be the frame over PG such that

- $W^{i+1} = W^i \setminus (W_{\rightarrow}^i \cup W_{\leftarrow}^i)$,
- R^{i+1} is the function from $W^{i+1}{}^2$ to 2^{PG} such that
 - for all $a \in PG$, for all $x^{i+1}, y^{i+1} \in W^{i+1}$, $x^{i+1} R_a^{i+1} y^{i+1}$ iff for all $x \in W$, for all $x', y' \in W'$, if $x' = y'$ and $(x, x') \in x^{i+1}$ then there exists $y \in W$ such that $x R_a y$ and $(y, y') \in y^{i+1}$,

$Z^{i+1} \subseteq W \times (W' \times W^{i+1})$ be the binary relation such that for all $x \in W$, for all $(x', x^{i+1}) \in W' \times W^{i+1}$, $x Z^{i+1} (x', x^{i+1})$ iff $(x, x') \in x^{i+1}$.

Lemma 5.8 For all $i \geq 0$,

- $W^{fff} \subseteq W^i$,
- for all $x^{fff}, y^{fff} \in W^{fff}$, $R^{fff}(x^{fff}, y^{fff}) \subseteq R^i(x^{fff}, y^{fff})$,
- $Z^{fff} \subseteq Z^i$.

Proof. The proof is similar to the proof of lemma 4.8. □

It follows that there exists $i_0 \geq 0$ such that

Lemma 5.9 $Z^{i_0}: \mathcal{F} \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^{i_0}$.

Proof. The proof is similar to the proof of lemma 4.9. □

Hence,

Lemma 5.10 $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^{i_0}, (x'_0, f(x''_0))$.

The above construction has the following property. When applied to an arbitrary finite set PG of program variables, arbitrary finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}' = (W', R')$ over PG , arbitrary roots $x_0 \in W$, $x'_0 \in W'$, it stops with a frame $\mathcal{F}^{i_0} = (W^{i_0}, R^{i_0})$ over PG and a binary relation $Z^{i_0} \subseteq W \times (W' \times W^{i_0})$ such that if $W^{i_0} \neq \emptyset$ then $Z^{i_0}: \mathcal{F} \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^{i_0}$. Hence, we can give a simple algorithm for solving (SO):

- For all $i \geq 0$, construct the frame $\mathcal{F}^i = (W^i, R^i)$ over PG and the binary relation $Z^i \subseteq W \times (W' \times W^i)$ as above until if $W^i \neq \emptyset$ then $Z^i: \mathcal{F} \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}^i$.

Not surprisingly, the above algorithm returns the value *true* iff there exists a frame $\mathcal{F}'' = (W'', R'')$ over PG , a root $x''_0 \in W''$ such that $\mathcal{F}, x_0 \longleftrightarrow \mathcal{F}' \oplus \mathcal{F}'', (x'_0, x''_0)$.

Seeing that \mathcal{F}^0 and Z^0 can be constructed in exponential time and for all $i \geq 0$, \mathcal{F}^{i+1} and Z^{i+1} can be constructed in time polynomial in the size of \mathcal{F}^i and Z^i , it follows immediately that

Proposition 5.11 *(SO) is in EXPTIME.*

6 Conclusion

We have considered the decision problems (SC) and (SO) of controller/orchestrator synthesis. Deterministic algorithms that check in exponential-time whether a controller/orchestrator exists have been proposed. An interesting (and still open) question is to evaluate the exact complexity of (SC) and (SO). Let us remark that the following decision problem is known to be EXPTIME-hard: given a finite set PG of program variables, deterministic finite frames $\mathcal{F} = (W, R)$, $\mathcal{F}^1 = (W^1, R^1)$, ..., $\mathcal{F}^n = (W^n, R^n)$ over PG , roots $x_0 \in W$, $x^1_0 \in W^1$, ..., $x^n_0 \in W^n$, determine if \mathcal{F}, x_0 is simulated by $\mathcal{F}_1 \oplus \dots \oplus \mathcal{F}_n, (x^1_0, \dots, x^n_0)$. See [12] for details. Are (SC) and (SO) EXPTIME-hard too? If (SC) and (SO) prove to be EXPTIME-hard too then we doubt the practicality of any decision method for them. In this respect, the use of symbolic techniques should permit to reduce the practical cost of controller/orchestrator synthesis. Possible solutions would demand to use compact data structures for the representation of frames [8] and to apply the techniques of abstraction and refinement used within the context of computer-aided verification [9]. Variants of (SC) and (SO) can be considered as well. For instance, one may consider that the controller/orchestrator must be transitive, reverse well-founded, etc. For such a variant, although we believe that our filtration approach can provide a solution, the complexity of controller/orchestrator synthesis is still unknown. Take another variant: one may replace “bisimilar” by “trace equivalent”. For such a variant, although Ramadge and Wonham [14] and Tsitsiklis [15] have indirectly and partially addressed it, the complexity of controller/orchestrator synthesis is still unknown. Finally, one may involve atomic propositions and do everything on the level of finite models which are a more natural framework for the synthesis problems. Involving atomic propositions can make the synthesis problems much harder, at least in some cases. For instance, every two finite serial frames are bisimilar, hence the synthesis problems in the case of serial frames are trivial; but not so for serial models.

Acknowledgements

We make a point of thanking the colleagues of the *Institut de recherche en informatique de Toulouse* who, by the discussions we had with them, contributed to the development of the work we present today. Special acknowledgement is heartily granted as well to the anonymous referees who made several helpful comments for improving the readability of the paper and suggesting variants of (SC) and (SO).

References

- [1] Alvarez, C., Balcázar, J., Gabarró, J., Sántha, M.: *Parallel complexity in the design and analysis of concurrent systems*. In: *PARLE '91, Parallel Architectures and Languages Europe*. Springer-Verlag (1991) 288–303.
- [2] Arnold, A., Niwiński, D.: *Rudiments of μ -Calculus*. Elsevier (2001).
- [3] Arnold, A., Vincent, A., Walukiewicz, I.: *Games for synthesis of controllers with partial observation*. Theoretical computer science **303** (2003) 7–34.
- [4] Berardi, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Mecella, M.: *Automatic composition of e-services that export their behavior*. In: *Service-Oriented Computing — ICSOC 2003*. Springer (2003) 43–58.
- [5] Berardi, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Mecella, M.: *Synthesis of underspecified composite e-services based on automated reasoning*. In: *ICSOC'04 — Proceedings of the Second International Conference on Service Oriented Computing*. ACM Press (2004) 105–114.
- [6] Berardi, D., Calvanese, D., De Giacomo, G., Hull, R., Mecella, M.: *Automated composition of transition-based semantic Web services with messaging*. In: *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB Endowment (2005) 613–624.
- [7] Blackburn, P., de Rijke, M., Venema, Y.: *Modal Logic*. Cambridge University Press (2001).
- [8] Burch, J., Clarke, E., McMillan, K., Dill, D., Hwang, L.: *Symbolic model checking: 10^{20} states and beyond*. Information and Computation **98** (1992) 142–170.
- [9] Clarke, E., Grumberg, O., Jha, S., Lu, Y., Veith, H.: *Counterexample-guided abstraction refinement for symbolic model checking*. Journal of the ACM **50** (2003) 752–794.
- [10] Harel, D., Kozen, D., Tiuryn, J.: *Dynamic Logic*. MIT Press (2000).
- [11] Maler, O., Pnueli, A., Sifakis, J.: *On the synthesis of discrete controllers for timed systems*. In: *STACS 95*. Springer (1995) 229–242.
- [12] Muscholl, A., Walukiewicz, I.: *A lower bound on Web services composition*. In: *Foundations of Software Science and Computational Structures*. Springer (2007) 274–286.
- [13] Pistore, M., Marconi, A., Bertoli, P., Traverso, P.: *Automated composition of Web services by planning at the knowledge level*. In: *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence (2005) 1252–1259.
- [14] Ramadge, P., Wonham, W.: *The control of discrete event systems*. In: *Proceedings of the IEEE*. IEEE (1989) 81–98.
- [15] Tsitsiklis, J.: *On the control of discrete-event dynamical systems*. Mathematics of Control, Signals, and Systems **2** (1989) 95–107.