BRICS

Basic Research in Computer Science

BRICS RS-99-26 J. G. Henriksen: An Expressive Extension of TLC

An Expressive Extension of TLC

Jesper G. Henriksen

BRICS Report Series RS-99-26

ISSN 0909-0878 September 1999

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone:	**+45 8942 3360**
Telefax:	**+45 8942 3255**
Internet:	**BRICS@brics.dk**

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/26/`

An Expressive Extension of TLC

Jesper Gulmann Henriksen[*]
BRICS[†], Department of Computer Science,
University of Aarhus, Denmark
gulmann@brics.dk

September 27, 1999

Abstract

A temporal logic of causality (TLC) was introduced by Alur, Penczek and Peled in [1]. It is basically a linear time temporal logic interpreted over Mazurkiewicz traces which allows quantification over causal chains. Through this device one can directly formulate causality properties of distributed systems. In this paper we consider an extension of TLC by strengthening the chain quantification operators. We show that our logic TLC* adds to the expressive power of TLC. We do so by defining an Ehrenfeucht-Fraïssé game to capture the expressive power of TLC. We then exhibit a property and by means of this game prove that the chosen property is not definable in TLC. We then show that the same property is definable in TLC*. We prove in fact the stronger result that TLC* is expressively stronger than TLC exactly when the dependency relation associated with the underlying trace alphabet is not transitive.

1 Introduction

One traditional approach to automatic program verification is model checking LTL [13] specifications. In this context, the model checking problem is to

[*] Part of this work was done at Lehrstuhl für Informatik VII, RWTH Aachen, Germany
[†] **B**asic **R**esearch **i**n **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

decide whether or not all computation sequences of the system at hand satisfy the required properties formulated as an assertion of LTL. Several software packages exploiting the rich theory of LTL are now available to carry out the automated verification task for quite large finite-state systems, e.g. [2, 9].

Usually computations of a distributed system will constitute interleavings of the occurrences of causally independent actions. Often, the computation sequences can be naturally grouped together into equivalence classes of sequences corresponding to different interleavings of the same partially ordered computation stretch. For a large class of interesting properties expressed by linear time temporal logics, it turns out that either all members of an equivalence class satisfy a certain property or none do. For such properties the computional resources needed for the verification task can be substantially reduced by means of the so-called partial-order methods for verification [8, 12, 18].

Such equivalence classes can be canonically represented by restricted labelled partial orders known as Mazurkiewicz traces [5, 10]. These objects — apart from alleviating the state-explosion problems of verification — also allow direct formulations of properties expressing concurrency and causality. A number of linear time temporal logics to be interpreted directly over Mazurkiewicz traces (e.g. [1, 4, 11, 14, 15, 16, 17]) has been proposed in the literature starting with TrPTL [15].

Among these, we consider here a temporal logic of causality (TLC) introduced in [1] to express serializability (of partially ordered computations) in a direct fashion. The operators of TLC are essentially the branching-time operators of CTL [3] interpreted over causal chains of traces. However, the expressive power of this logic has remained an interesting open problem. Indeed, not much is known about the relative expressive powers of the various temporal logics over traces.

What is known is that a linear time temporal LTrL, patterned after LTL, was introduced [17] and proven expressively equivalent to the first-order theory of traces [6]. LTrL has a simple and natural formulation with very restricted past operators, but was shown non-elementary in [19]. Recently, it was shown that the restricted past operators of LTrL can be replaced by certain new future operators while maintaining expressive completeness. In other work, Niebert introduced a fixed point based linear time temporal logic [11]. This logic has an elementary-time decision procesure and is equal in expressive power to the monadic second-order theory of traces.

However, the expressive powers of most other logics put forth (e.g. [1, 14, 15]) still have an unresolved relationship to each other and, in particular, to first-order logic. Most notably, it is still a challenging open problem whether or not TrPTL or TLC is expressively weaker than first-order logic. With virtually no other seperation result known, this paper is a contribution towards understanding the relative expressive power of such logics.

A weakness of TLC is that it doesn't facilitate direct reasoning about causal relationships between the individual events on the causal chains. In this paper we remedy this deficiency and extend TLC by strengthening quantification over causal chains. This extended logic, which we call TLC*, will enjoy a similarity to CTL* [3] that TLC has to CTL. The main result of this paper is that our extension TLC* is expressively stronger than TLC for general trace alphabets whereas they express the same class of properties over trace alphabets with a transitive dependency relation. We prove this result with the aid of an Ehrenfeucht-Fraïssé game for traces that we develop. To our knowledge this is the first instance of the use of such games to obtain seperation results for temporal logics defined over partial orders. We believe that this approach is fruitful and that similar techniques may lead to other seperation results within this area.

In the next section we briefly recall Mazurkiewicz traces and a few related notions. In Section 3 we introduce TLC and TLC*, the main objects of study in this paper. We give a very simple and natural example of a property easily captured in TLC* but not in TLC. In Section 4 we define an Ehrenfeucht-Fraïssé game and prove its correspondence to TLC. We use this correspondence in Section 5 to exhibit a property which we prove is undefinable in TLC. In Section 6 we show that the said property can be defined within TLC*. Finally, we put all the pieces together to arrive at the main result. We conclude by a quick overview of the relative expressive powers of logics over traces.

2 Preliminaries

A *(Mazurkiewicz) trace alphabet* is a pair (Σ, I), where Σ, the alphabet, is a finite set and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric *independence relation.* Usually, Σ consists of the actions performed by a distributed system while I captures a static notion of causal independence between actions. For the rest of the section we fix a trace alphabet (Σ, I). We define $D = (\Sigma \times \Sigma) - I$ to be the *dependency relation* which is then reflexive and symmetric.

Let $T = (E, \leq, \lambda)$ be a Σ-labelled poset. In other words, $(E, \leq)$ is a poset and $\lambda : E \to \Sigma$ is a labelling function. For $e \in E$ we define $\downarrow e = \{x \in E \mid x \leq e\}$. We also let $\lessdot$ be the *covering relation* given by $x \lessdot y$ iff $x < y$ and for all $z \in E$, $x \leq z \leq y$ implies $x = z$ or $z = y$. Moreover, we let the *concurrency relation* be defined as $x \; co \; y$ iff $x \not\leq y$ and $y \not\leq x$. A *Mazurkiewicz trace* (over (Σ, I)) is then a Σ-labelled poset $T = (E, \leq, \lambda)$ satisfying:

(T1) $\forall e \in E.$ $\quad \downarrow e$ is a finite set
(T2) $\forall e, e' \in E.$ $\quad e \lessdot e'$ implies $\lambda(e) \; D \; \lambda(e')$.
(T3) $\forall e, e' \in E.$ $\quad \lambda(e) \; D \; \lambda(e')$ implies $e \leq e'$ or $e' \leq e$.

We shall let $TR(\Sigma, I)$ denote the class of traces over (Σ, I). As usual, a trace language L is a subset of traces, i.e. $L \subseteq TR(\Sigma, I)$. Throughout the paper we will not distinguish between isomorphic elements in $TR(\Sigma, I)$. We will refer to members of E as *events*. It will be convenient to assume the existence of a unique least event $\perp \in E$ corresponding to a system initialization event carrying no label, i.e. $\lambda(\perp)$ is undefined and $\perp < e$ for every $e \in E - \{\perp\}$.

It's not hard to show that the traces introduced above as restricted labelled partial orders can be equivalently represented as congruence classes of strings corresponding to the same partially ordered computations (and vice versa; see e.g. [16]). We will sometimes abuse notation and let a string in Σ^* denote its corresponding trace in (Σ, I) whenever no confusion arises. This is enforced by using conventional parentheses for string languages and square brackets for trace languages.

In setting the scene for defining the semantics of formulas of TLC* we first introduce some notation for sequences. The length of a finite sequence ρ will be denoted by $|\rho|$. In case ρ is infinite we set $|\rho| = \omega$. Let $\rho = (e_0, e_1, \ldots, e_n, \ldots)$ and $0 \leq k < |\rho|$. We set $\rho_k = (e_k, e_{k+1}, \ldots, e_n, \ldots)$.

Let $T = (E, \leq, \lambda)$ be a trace over (Σ, I). A *future causal chain rooted at* $e \in E$ is a (finite or infinite) sequence $\rho = (e_0, e_1, \ldots, e_n, \ldots)$ with $e = e_0$, $e_i \in E$ such that $e_{i-1} \lessdot e_i$ for every $i \geq 1$. The labelling function $\lambda : E \to \Sigma$ is extended to causal chains in the obvious way by: $\lambda(\rho) = (\lambda(e_0)\lambda(e_1) \cdots \lambda(e_n) \cdots)$. We say that a future causal chain ρ is *maximal* in case ρ is either infinite or it is finite and there exists no $e' \in E$ such that $e_{|\rho|} \lessdot e'$. A *past causal chain rooted at* $e \in E$ is a (finite) sequence $\rho = (e_n, \ldots, e_1, e_0)$ with $e = e_0$, $e_i \in E$ such that $e_i \lessdot e_{i-1}$ for every $1 \leq i \leq n$.

3 Syntax and Semantics

In this section we will define the syntax and semantics of the temporal logics over traces to be considered in this paper. We start by introducing TLC* and continue by giving an explicit definition of the sublogic TLC. We will not define first-order logic over traces (FO), but we refer the reader to e.g. [6, 16, 17].

TLC* consists of three different syntactic entities; event formulas (Φ_{ev}), future chain formulas (Φ^+_{ch}) and past chain formulas (Φ^-_{ch}) defined by mutual induction as described below:

$$\begin{array}{lcl} \Phi_{ev} & ::= & p_a \mid {\sim}\alpha \mid \alpha_1 \vee \alpha_2 \mid \mathrm{co}(\alpha) \mid E(\phi) \mid E^-(\psi),\ \text{ with } a \in \Sigma. \\ \Phi^+_{ch} & ::= & \alpha \mid {\sim}\phi \mid \phi_1 \vee \phi_2 \mid X\phi \mid \phi_1 U \phi_2. \\ \Phi^-_{ch} & ::= & \alpha \mid {\sim}\psi \mid \psi_1 \vee \psi_2 \mid X^-\psi \mid \psi_1 U^- \psi_2\ , \end{array}$$

where α, ϕ and ψ with or without subscripts here and throughout the rest of the paper are formulas of Φ_{ev}, Φ^+_{ch} and Φ^-_{ch}, respectively. The formulas of TLC$^*(\Sigma, I)$ are the set of *event* formulas Φ_{ev} as defined above[1].

The semantics of formulas of TLC* is divided into two parts; event formulas and chain formulas. Let $T \in TR(\Sigma, I)$ and $e \in E$. The notion of an event formula α being satified at an event e of T is defined inductively in the following manner.

- $T, e \models p_a$ iff $\lambda(e) = a$.
- $T, e \models {\sim}\alpha$ iff $T, e \not\models \alpha$.
- $T, e \models \alpha_1 \vee \alpha_2$ iff $T, e \models \alpha_1$ or $T, e \models \alpha_2$.
- $T, e \models \mathrm{co}(\alpha)$ iff there exists an $e' \in E$ with $e\ co\ e'$ and $T, e' \models \alpha$.
- $T, e \models E(\phi)$ iff there exists a future causal chain ρ rooted at e with $T, \rho \models \phi$.
- $T, e \models E^-(\psi)$ iff there exists a past causal chain ρ rooted at e with $T, \rho \models \psi$.

[1] Another logic was in [1] termed "TLC*", but as that logic denoted TLC interpreted over linearizations it is unrelated to our logic which seems naturally to earn the name "TLC*".

As usual, $\mathrm{tt} = p_a \vee \sim p_a$ and $\mathrm{ff} = \sim\mathrm{tt}$. Suppose $\rho = (e_0, e_1, \ldots, e_n, \ldots)$ is a future causal chain. The notion of $T, \rho \models \phi$ for a future chain formula ϕ is defined inductively below.

- $T, \rho \models \alpha$ iff $T, e_0 \models \alpha$.
- $T, \rho \models \sim\phi$ iff $T, \rho \not\models \phi$.
- $T, \rho \models \phi_1 \vee \phi_2$ iff $T, \rho \models \phi_1$ or $T, \rho \models \phi_2$.
- $T, \rho \models X\phi$ iff $T, \rho_1 \models \phi$.
- $T, \rho \models \phi_1 U \phi_2$ iff there exists a $0 \leq k < |\rho|$ such that $T, \rho_k \models \phi_2$. Moreover, $T, \rho_m \models \phi_1$ for each $0 \leq m < k$.

The notion of $T, \rho \models \psi$ for a past causal chain ρ and past chain formula ψ is defined in the straightforward manner. The well-known future chain operators are derived as $F\phi = \mathrm{tt}U\phi$ and $G\phi = \sim F \sim \phi$.

Suppose $T \in TR(\Sigma, I)$ and $\alpha \in \mathrm{TLC}^*(\Sigma, I)$. Then T *satisfies* α iff $T, \perp \models \alpha$, denoted $T \models \alpha$. The *language defined by* α is: $\mathcal{L}(\alpha) = \{T \in TR(\Sigma, I) \mid T \models \alpha\}$. We say that $L \subseteq TR(\Sigma, I)$ is *definable* in TLC* if there exists some $\alpha \in \mathrm{TLC}^*(\Sigma, I)$ such that $\mathcal{L}(\alpha) = L$. By slight abuse of notation, the class of trace languages over (Σ, I) definable in TLC* will also be denoted by $\mathrm{TLC}^*(\Sigma, I)$.

The formulas of $\mathrm{TLC}(\Sigma, I)$ — introduced in [1] with a slightly different syntax — is then the set of formulas of $\mathrm{TLC}^*(\Sigma, I)$ where each of the chain operators X, U, G, X^-, U^- is immediately preceded by a chain quantifier E. As TLC will play a prominent role in this paper we will bring out its definition in more detail. More precisely, the set of formulas is given as:

$$\begin{aligned} \mathrm{TLC}(\Sigma, I) \ ::= \ & p_a \mid \sim\alpha \mid \alpha \vee \beta \mid EX(\alpha) \mid EU(\alpha, \beta) \mid \\ & EG(\alpha) \mid EX^-(\alpha) \mid EU^-(\alpha, \beta) \mid \mathrm{co}(\alpha), \end{aligned}$$

where $a \in \Sigma$. The semantics is inherited directly from TLC* in the obvious manner, so notions of definability etc. are carried over directly. It can be shown that our extension TLC* remains decidable. In work to appear we construct an elementary-time decision procedure for TLC* by means of Büchi automata.

Hence while the formulas of TLC are basically the well-known operators of the branching-time logic CTL [3] augmented with symmetrical past operators and concurrency information, the operators of TLC* are basically the well-known operators of CTL*[3] similarly extended with past quantifiers in a restricted fashion as well as concurrency information. The crucial difference is that while CTL and CTL* are branching-time logics interpreted over Kripke structures, TLC and TLC* are linear time temporal logics on traces interpreted over the underlying Hasse diagrams of the partial orders.

One of the weaknesses of TLC is that it doesn't directly facilitate reasoning about causal relationships of the individual events of the causal chains at hand. As a consequence, a number of interesting properties are not (either easily or at all) expressible within TLC. Section 5 provides a formal proof of this claim, but we will in the following bring out another such property which is very natural.

Suppose that a and b are actions representing the acquiring and releasing, respectively, of some resource. A relevant property of this system is then whether or not there exists some causal chain in the execution of the system — presumably containing other system actions than $\{a, b\}$ — such that the a's and b's alternate strictly until the task is perhaps eventually completed. Via the future chain formula $\phi_{xy} = p_x \to X(\sim(p_x \vee p_y)U(p_y))$ we can easily express this property in TLC* by $E(G(\phi_{ab} \wedge \phi_{ba}))$. The point is here that TLC* allows us to investigate each causal chain in mention by a causal chain formula, which is then confined to this very chain. This is not possible in TLC, as the existential quantifications interpreted at some fixed event of the chain would potentially consider *all* causal chains originating at this event — not just the one presently being investigated.

We conclude with two important notions relating to TLC. Firstly, let α be a formula of TLC(Σ, I). The *operator depth* of α is defined inductively as follows:

- $od(p_a) = 0$.
- $od(\sim\alpha) = od(\alpha)$ and $od(\alpha \vee \beta) = \max(od(\alpha), od(\beta))$.
- $od(EX(\alpha)) = od(EG(\alpha)) = od(EX^-(\alpha)) = od(\mathrm{co}(\alpha)) = 1 + od(\alpha)$.
- $od(EU(\alpha, \beta)) = od(EU^-(\alpha, \beta)) = 1 + \max(od(\alpha), od(\beta))$.

The set of formulas of operator depth k is denoted by $OD(k)$.

Given $T_0, T_1 \in TR(\Sigma, I)$ and e_i events of T_i we define that $(T_0, e_0) \equiv_n (T_1, e_1)$ if for any formula $\alpha \in \mathrm{TLC}(\Sigma, I)$ with $od(\alpha) \leq n$, $T_0, e_0 \models \alpha$ iff $T_1, e_1 \models \alpha$, i.e. both structures agree on all subformulas of operator depth at most $n \geq 0$. It is then not hard to see that $(T_0, e_0) \equiv_0 (T_1, e_1)$ iff e_0 and e_1 are identically labelled, i.e. either $\lambda(e_0) = \lambda(e_1)$ or $e_0 = e_1 = \bot$.

4 An Ehrenfeucht-Fraïssé Game for TLC

In this section we will present an Ehrenfeucht-Fraïssé game to capture the expressive power of TLC. The game is played directly on the poset representation of (finite or infinite) Mazurkiewicz traces and it is similar in spirit to the Ehrenfeucht-Fraïssé game for LTL introduced by Etessami and Wilke [7]. We extend their approach to the richer setting of traces by highlighting current causal chains in the until-based moves and adding past- and co-moves.

The EF-TLC game is a game played between two persons, Spoiler and Preserver, on a pair of traces (T_0, T_1). The game is played over k rounds starting from an initial game state (e_0, e_1) and after each round the current game state is a pair of events (e'_0, e'_1) with $e'_i \in E_i$. Each round starts with the game in some specific initial game state (e_0, e_1) and Spoiler chooses one of the moves defined below and the game proceeds accordingly:

***EX*-Move:** This move can only be played by Spoiler if there exists an $e'_0 \in E_0$ such that $e_0 \lessdot e'_0$ or there exists an $e'_1 \in E_1$ such that $e_1 \lessdot e'_1$. Spoiler then wins the game in case there either exists no $e'_0 \in E_0$ such that $e_0 \lessdot e'_0$ or no $e'_1 \in E_1$ such that $e_1 \lessdot e'_1$. Otherwise (in which case both e_0 and e_1 has $\lessdot$-successors) the game proceeds as follows:

1. Spoiler chooses $i \in \{0, 1\}$, and an event $e'_i \in E_i$ such that $e_i \lessdot e'_i$.
2. Preserver responds by choosing an event $e'_{1-i} \in E_{1-i}$ such that $e_{1-i} \lessdot e'_{1-i}$.
3. The new game state is now (e'_0, e'_1).

EU-**Move:**

1. Spoiler chooses $i \in \{0,1\}$, and an event $e'_i \in E_i$ such that $e_i \leq e'_i$ and he highlights a future causal chain $(e_i = f_i^0, f_i^1, \ldots, f_i^n = e'_i)$ with $n \geq 0$.
2. Preserver responds by choosing an event $e'_{1-i} \in E_{1-i}$ with $e_{1-i} \leq e'_{1-i}$ such that if $e_i = e'_i$ then $e_{1-i} = e'_{1-i}$. Furthermore she highlights a future causal chain $(e_{1-i} = f_{1-i}^0, f_{1-i}^1, \ldots, f_{1-i}^m = e'_{1-i})$ with $m \geq 0$.
3. Spoiler now chooses one of the following two steps:
   - Spoiler sets the game state to (e'_0, e'_1).
   - Spoiler chooses an event $f_{1-i} \in \{f_{1-i}^0, f_{1-i}^1 \ldots f_{1-i}^m\}$. Preserver responds with an event $f_i \in \{f_i^0, f_i^1 \ldots f_i^n\}$ and the game continues in the state (f_0, f_1).

EG-**Move:**

1. Spoiler chooses $i \in \{0,1\}$, and highlights a maximal future causal chain $(e_i = f_i^0, f_i^1, \ldots, f_i^n, \ldots)$ with $f_i^j \in E_i$ and $n \geq 0$.
2. Preserver responds by highlighting a maximal future causal chain $(e_{1-i} = f_{1-i}^0, f_{1-i}^1, \ldots, f_{1-i}^m, \ldots)$ with $f_i^j \in E_{1-i}$ and $m \geq 0$.
3. Spoiler chooses an event $f_{1-i} \in \{f_{1-i}^0, f_{1-i}^1 \ldots f_{1-i}^m\}$. Preserver responds with an event $f_i \in \{f_i^0, f_i^1 \ldots f_i^n\}$ and the game continues in the state (f_0, f_1).

co-**Move:** This move can only be played by Spoiler if there exists an $e'_0 \in E_0$ such that e_0 *co* e'_0 or there exists an $e'_1 \in E_1$ such that e_1 *co* e'_1. Spoiler then wins the game in case there either exists no $e'_0 \in E_0$ such that e_0 *co* e'_0 or no $e'_1 \in E_1$ such that e_1 *co* e'_1. Otherwise (in which case both e_0 and e_1 have concurrent events) the game proceeds as follows:

1. Spoiler chooses $i \in \{0,1\}$, and an event $e'_i \in E_i$ such that e_i *co* e'_i in T_i.
2. Preserver responds by choosing an event $e'_{1-i} \in E_{1-i}$ such that e_{1-i} *co* e'_{1-i} in T_{1-i}.
3. The new game state is now (e'_0, e'_1).

EX^-**-Move:** This move can only be played by Spoiler if $e_0 \neq \bot$ or $e_1 \neq \bot$. Spoiler then wins the game in case $e_0 = \bot$ or $e_1 = \bot$. Otherwise (in which case neither e_0 nor e_1 is $\bot$) the game proceeds as follows:

1. Spoiler chooses $i \in \{0, 1\}$, and an event $e'_i \in E_i$ such that $e'_i \lessdot e_i$.
2. Preserver responds by choosing an event $e'_{1-i} \in E_{1-i}$ such that $e'_{1-i} \lessdot e_{1-i}$.
3. The new game state is now (e'_0, e'_1).

EU^-**-Move:**

1. Spoiler chooses $i \in \{0, 1\}$, and an event $e'_i \in E_i$ such that $e'_i \leq e_i$ and he highlights a past causal chain $(e'_i = f_i^n, f_i^{n-1}, \ldots, f_i^0 = e_i)$ with $n \geq 0$.
2. Preserver responds by choosing an event $e'_{1-i} \in E_{1-i}$ with $e'_{1-i} \leq e_{1-i}$ such that if $e_i = e'_i$ then $e_{1-i} = e'_{1-i}$. Furthermore she highlights a past causal chain $(e'_{1-i} = f_{1-i}^m, f_{1-i}^{m-1}, \ldots, f_{1-i}^0 = e_{1-i})$ with $m \geq 0$.
3. Spoiler now chooses one of the following two steps:
   - Spoiler sets the game state to (e'_0, e'_1).
   - Spoiler chooses an event $f_{1-i} \in \{f_{1-i}^0, f_{1-i}^1 \ldots f_{1-i}^m\}$. Preserver responds with an event $f_i \in \{f_i^0, f_i^1 \ldots f_i^n\}$ and the game continues in the state (f_0, f_1).

In the 0-round game Spoiler wins if $(T_0, e_0) \not\equiv_0 (T_1, e_1)$ and otherwise Preserver wins. In the $(k+1)$-round game Spoiler wins if $(T_0, e_0) \not\equiv_0 (T_1, e_1)$. If it is the case that $(T_0, e_0) \equiv_0 (T_1, e_1)$, a round is played according to the above moves. This round either results in a win for Spoiler (e.g. by the EX-move) or a new game state (e'_0, e'_1). In the latter case, a k-round game is then played starting from the initial game state (e'_0, e'_1).

We say that *Preserver has a winning strategy* in the k-round game on (T_0, e_0) and (T_1, e_1), denoted $(T_0, e_0) \sim_k (T_1, e_1)$, if she can win the k-round game on the structures T_0 and T_1 starting in the initial game state (e_0, e_1) no matter which moves are performed by Spoiler. If not, we say that *Spoiler has a winning strategy.* We refer to [7] for basic intuitions about the game.

Our interest in the game lies in the following fact.

Proposition 4.1 *For every $k \geq 0$, $(T_0, e_0) \sim_k (T_1, e_1)$ iff $(T_0, e_0) \equiv_k (T_1, e_1)$.*

Proof: We prove that $(T_0, e_0) \sim_k (T_1, e_1)$ iff $(T_0, e_0) \equiv_k (T_1, e_1)$ by induction on k. The base case where $k = 0$ follows trivially from the definition.

For the inductive step suppose that the claim is true for k. We first prove the direction from left to right. Suppose that $(T_0, e_0){\sim_{k+1}}(T_1, e_1)$. Let $\alpha \in \mathrm{TLC}(\Sigma, I)$ with $od(\alpha) = k + 1$. We must show that $T_0, e_0 \models \alpha$ iff $T_1, e_1 \models \alpha$. It suffices to prove the statement when the top-level connective of α is a chain-operator because by boolean combinations (T_0, e_0) and (T_1, e_1) would then agree on all formulas of operator depth $k + 1$. We will only consider the case where the top-level chain-operator is EU. The other cases follow similarly.

Suppose now $\alpha = EU(\beta, \beta')$. Assume without loss of generality that $T_0, e_0 \models \alpha$, i.e. there exists a future causal chain $\rho^0 = (f_0^0, f_0^1, \ldots, f_0^n)$ with $e_0 = f_0^0$ and $f_0^n = e'_0$ such that $T_0, f_0^j \models \beta$ for each $0 \leq j < n$ and $T_0, e'_0 \models \beta'$. Hence we let Spoiler play the EU-move on T_0 and make him highlight ρ^0 on T_0. Preserver now uses her winning strategy and highlights $\rho^1 = (f_1^0, f_1^1, \ldots, f_1^m)$ with $e_1 = f_1^0$ and $f_1^m = e'_1$. Two subcases now arise.

Assume first that Spoiler sets the new game state to (e'_0, e'_1). As e'_1 was chosen from Preserver's winning strategy we have that $(T_0, e'_0){\sim_k}(T_1, e'_1)$ which by induction hypothesis implies that $(T_0, e'_0){\equiv_k}(T_1, e'_1)$. Thus $T_1, e'_1 \models \beta'$. Now, assume that Spoiler instead picked an event f_1 on ρ^1. By Preserver's winning strategy she could pick an event f_0 on ρ^0 (This is possible due to the requirement that if $e_0 = e'_0$ then $e_1 = e'_1$). Again by the winning strategy we have that $(T_0, f_0){\sim_k}(T_1, f_1)$ and by induction hypothesis that $T_1, f_1 \models \beta$. Hence $T_1, f_1 \models EU(\beta, \beta')$, which concludes this direction of the proof.

We prove the direction from right to left by contraposition, so suppose that $(T_0, e_0) \not\sim_{k+1} (T_1, e_1)$. We will then exhibit a formula $\alpha \in \mathrm{TLC}(\Sigma, I)$ with $od(\alpha) = k + 1$ such that $T_0, e_0 \models \alpha$ but $T_1, e_1 \not\models \alpha$. Again, we will only prove the case where Spoiler's first move of his winning strategy is either the EU-move. The other cases either follows in analogous or easier manners.

Suppose Spoiler plays the EU-move on T_0 (without loss of generality), i.e. he chooses a future causal chain $\rho^0 = (f_0^0, f_0^1, \ldots, f_0^n)$ with $e_0 = f_0^0$ and $f_0^n = e'_0$. It is not hard to show by induction that there are only a finite number of semantically inequivalent formulas α with $od(\alpha) \leq k$ and $T_0, e \models \alpha$ for any $e \in E_0$. Hence, each formula $\beta_0^j = \bigwedge\{\alpha \in OD(k) \mid T_0, f_0^j \models \alpha\} \wedge \bigwedge\{\sim\alpha \in OD(k) \mid T_0, f_0^j \not\models \alpha\}$ is well-defined and equivalent to a formula of operator depth k for each $0 \leq j < n$, so letting $\beta_{e'_0} = \beta_0^n$ we have that $\alpha = EU(\bigvee_{0 \leq j < n} \beta_0^j, \beta_{e_0'})$ is a TLC-formula with $od(\alpha) = k + 1$ and by definition $T_0, e_0 \models \alpha$. We will argue that $T_1, e_1 \not\models \alpha$.

Suppose that $T_1, e_1 \models \alpha$. Then there exists a future causal chain $\rho^1 = (f_1^0, f_1^1, \ldots, f_1^m)$ with $e_1 = f_1^0$ and $f_1^m = e_1'$ such that $T_1, f_1^l \models \bigvee_{0 \leq j < n} \beta_0^j$ for each $0 \leq l < m$ and $T_1, e_1' \models \beta_{e_0'}$.

Assume first that Spoiler chooses to set the new game state to (e_0', e_1') by following his winning strategy. As $T_1, e_1' \models \beta_{e_0'}$ it must be the case that for each $\gamma \in OD(k)$, $T_0, e_0' \models \gamma$ iff $T_1, e_1' \models \gamma$. By induction hypothesis $(T_0, e_0') \sim_k (T_1, e_1')$ which contradicts that Spoiler has a winning strategy because Preserver could initially have played ρ^1 as above and continued according to $(T_0, e_0') \sim_k (T_1, e_1')$.

Now assume that Spoiler instead by his winning strategy picks an event f_1 on ρ^1. Then $T_1, f_1 \models \beta_0^j$ for some $0 \leq j < n$ as $T_1, f_1 \models \bigvee_{0 \leq j < n} \beta_0^j$. Again by induction hypothesis we know that $(T_0, f_0^j) \sim_k (T_1, f_1)$ which again contradicts that Spoiler has a winning strategy because Preserver could respond by picking $f_0^j \in E_0$ and continue from the game state (f_0^j, f_1) according to $(T_0, f_0^j) \sim_k (T_1, f_1)$.

Hence $T_1, e_1 \not\models \alpha$ as required. □

5 An Undefinability Result

In this section we will give an example of a natural property which we, by means of the game characterization of the previous section, will show is not definable in TLC. Let (Σ, I) be a trace alphabet with $\{a, b, c\} \subseteq \Sigma$ such that $a\ D\ c$ and $c\ D\ b$ but $a\ I\ b$. Consider $L = [abcabc]^* \subseteq TR(\Sigma, I)$.

Lemma 5.1 *L is not definable in* TLC(Σ, I).

Proof: Let $k \geq 0$ be given and consider $T_0^k = [abc]^{4k}$ and $T_1^k = [abc]^{4k+1}$. It suffices to show that $(T_0^k, \perp) \sim_k (T_1^k, \perp)$. By Proposition 4.1 it then follows that $(T_0^k, \perp) \equiv_k (T_1^k, \perp)$. Suppose L would be definable by a TLC-formula α of operator depth n. In particular then $(T_0^n, \perp) \equiv_n (T_1^n, \perp)$. However, by definition it must be the case that $T_0^n \in L$ and $T_1^n \notin L$, contradicting that T_0^n and T_1^n satisfy the same set of formulas of operator depth at most n. Hence, L cannot be expressed by any formula of TLC assuming $(T_0^k, \perp) \sim_k (T_1^k, \perp)$ holds for any $k \geq 0$. The remainder of the proof will be devoted to showing that it is the case that $(T_0^k, \perp) \sim_k (T_1^k, \perp)$. To bring this out we need a few definitions.

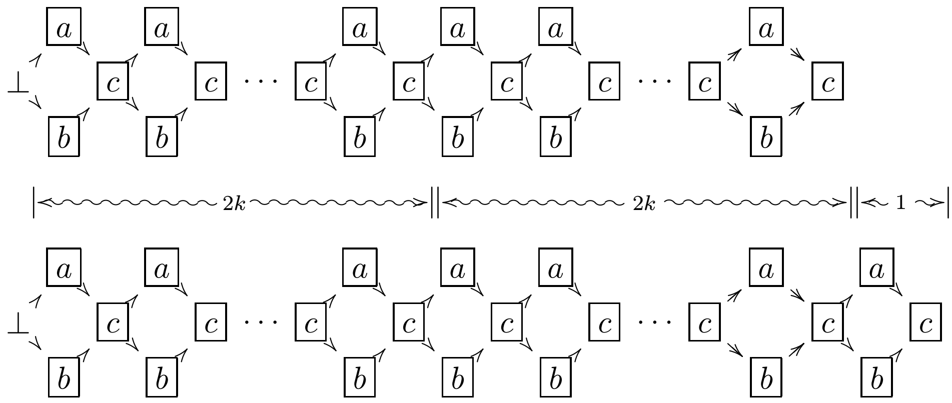

Figure 1: T_0^k (top) and T_1^k (bottom) on which the game is played.

As depicted in Figure 1 the game is played on T_0^k and T_1^k consisting of $4k$ and $4k+1$ copies of the trace factor $[abc]$, respectively. The section of e_i for $i \in \{0,1\}$ is then defined to be the number of the enclosing $[abc]$-factor in T_k^i counting from left and starting with 1. We denote this number by $sect(e_i)$. In case $e_i = \perp$ we set $sect(e_i) = 0$. Furthermore, we say that e_0 and e_1 are *position equivalent*, in case either $(e_0, e_1) = (\perp, \perp)$ or $\lambda(e_0) = \lambda(e_1)$. From the definition of T_0 and T_1 it follows that e_0 and e_1 are position equivalent in case e_0 and e_1 denote the same local positions in two (possibly distinct) sections of T_0 and T_1, respectively. The unique event of section $s \geq 1$ labelled with letter $x \in \{a,b,c\}$ in T_i^k will be denoted $e_i^{x,s}$. For example, the fourth b-labelled event of T_0^k is denoted $e_0^{b,4}$.

We will then show that Preserver has a strategy such that after $k' \leq k$ rounds played on (T_0^k, T_1^k) with current game state (e_0, e_1), the following invariant holds:

(i) e_0 and e_1 are position equivalent.

(ii) $sect(e_0) = sect(e_1)$ or $sect(e_0) = sect(e_1) - 1$.

(iii) $sect(e_0) = sect(e_1)$ implies $sect(e_0) \leq 2(k+k')$.

(iv) $sect(e_0) = sect(e_1) - 1$ implies $sect(e_0) \geq 2(k-k') + 1$.

We prove that the invariant holds by induction on k'. It is trivial to observe, that in the base case we have that $(e_0, e_1) = (\bot, \bot)$, $sect(e_0) = sect(e_1) = 0$ and $k' = 0$ thus satisfying (i),(ii), (iii) and (iv) above.

For the inductive step, assume that the statement holds for $k' < k$. From (i) it follows that $(T_0, e_0) \equiv_0 (T_1, e_1)$, so a next round is played. We then show that the Preserver can move so as to maintain the invariant for the next game state (e'_0, e'_1) by case analysis on the next move chosen by Spoiler. We only consider the case for the *EU*-move. The other moves follow analogously. From (ii) we know that $sect(e_0) = sect(e_1)$ or $sect(e_0) = sect(e_1) - 1$, so two subcases arise. Subcase I: $sect(e_0) = sect(e_1)$. Suppose Spoiler chooses to play the *EU*-move on T_0^k and highlights a future causal chain $\rho^0 = (e_0 = e_0^{x_0,s_0}, e_0^{x_1,s_1}, \ldots, e_0^{x_n,s_n} = e'_0)$. By assumption $sect(e_0) \leq 2(k + k')$.

Suppose first that $s_n \leq 2(k + k' + 1)$. Then Preserver can just copy the move and respond with $\rho^1 = (e_1 = e_1^{x_0,s_0}, e_1^{x_1,s_1}, \ldots, e_1^{x_n,s_n} = e'_1)$. If Spoiler chooses to set the new game state to (e'_0, e'_1), $sect(e'_0) = sect(e'_1) \leq 2(k+k'+1)$ and the invariant is maintained. If Spoiler instead chooses to pick an event $e_1^{x_i,s_i}$, Preserver would respond by picking $e_0^{x_i,s_i}$ and the invariant is maintained in a similar manner.

Suppose then that $s_n > 2(k + k' + 1)$. Preserver must then "insert" an additional occurrence of a section into ρ^0 at section $2(k + k' + 1)$. To bring this out, let l be the least index such that $s_l = 2(k + k' + 1)$, which exists by assumption. Preserver then responds with

$$\rho^1 = (e_1^{x_0,s_0}, \ldots, e_1^{x_l,s_l}, e_1^{x_{l+1},s_{l+1}}, e_1^{x_l,s_l+1}, e_1^{x_{l+1},s_{l+1}+1}, e_1^{x_{l+2},s_{l+2}+1}, \ldots, e_1^{x_n,s_n+1})$$

with $e'_1 = e_1^{x_n,s_n+1}$. If Spoiler chooses to set the new game state to (e'_0, e'_1), $sect(e'_0) = s_n = sect(e'_1) - 1$. However, the invariant is maintained as $sect(e'_0) \geq 2(k + k' + 1) \geq 2(k - (k' + 1)) + 1$. If Spoiler instead chooses to pick an event on ρ^1, Preserver responds dependent upon its index. If Spoiler picks one of the first $l + 2$ events $e_1^{x_i,s_i}$, Preserver responds with $e_0^{x_i,s_i}$. As $sect(e_0^{x_i,s_i}) = s_i = sect(e_1^{x_i,s_i}) \leq 2(k + k' + 1)$ the invariant is maintained. If Spoiler picks one of the remaining events $e_1^{x_i,s_i+1}$, Preserver responds with $e_0^{x_i,s_i}$ in which case $sect(e_0^{x_i,s_i}) = s_i = sect(e_1^{x_i,s_i+1}) - 1$ and the invariant is maintained as $sect(e_0^{x_i,s_i}) \geq 2(k + k' + 1) > 2(k - (k' + 1)) + 1$.

Suppose spoiler chooses to play the *EU*-move on T_1^k and highlights a future causal chain $\rho^1 = (e_1 = e_1^{x_0,s_0}, e_1^{x_1,s_1}, \ldots, e_1^{x_n,s_n} = e'_1)$. By assumption $sect(e_0) \leq 2(k + k')$. If $s_n \leq 2(k + k' + 1)$ then Preserver can, as above, just copy the move and maintain the invariant, so suppose that $s_n > 2(k+k'+1)$.

Preserver must then "chop" a duplicate occurrence off ρ^1 around the sections $2(k+k')+1, 2(k+k')+2 = 2(k+k'+1), 2(k+k')+3$ which exist by construction. Any causal chain passing through these three sections must pass (at least) two identical ac-labelled or bc-labelled stretches. Now, let l be the least index such that $s_l = 2(k+k')+1$ and consider the sequence $\sigma = (x_l, x_{l+2}, x_{l+4})$ with $\lambda(\sigma) \in \{a,b\}^3$. Remove from σ the first occurrence x_i where there exists an $j > i$ with x_j in σ and $x_i = x_j$. Let $\sigma' = (x_p, x_q)$ denote the resulting sequence where $p, q \in \{l, l+2, l+4\}$. Preserver then plays the chain ρ^0:

$$(e_0^{x_0,s_0}, \ldots, e_0^{x_{l-1},s_{l-1}}, e_0^{x_p,s_l}, e_0^{c,s_{l+1}}, e_0^{x_q,s_{l+2}}, e_0^{c,s_{l+3}}, e_0^{x_{l+5},s_{l+5}-1}, \ldots, e_0^{x_n,s_n-1})$$

with $e'_0 = e_0^{x_n,s_n-1}$. If Spoiler chooses to set the new game state to (e'_0, e'_1) then $\mathit{sect}(e'_0) = s_n = \mathit{sect}(e'_1) - 1$ so the invariant is maintained because $s_n > 2(k+k'+1) > 2(k-(k'+1))+1$. If Spoiler chooses to pick an event on ρ^0, Preserver responds according to one of several cases. If Spoiler picks one of the first l events $e_0^{x_i,s_i}$ then Preserver picks $e_1^{x_i,s_i}$ and the invariant is maintained as usual. If Spoiler picks either $e_0^{c,s_{l+1}}$ or $e_0^{c,s_{l+3}}$ then Preserver picks either $e_1^{c,s_{l+1}}$ or $e_1^{c,s_{l+3}}$, respectively. As the sections are both s_{l+1} or both s_{l+3} and $s_{l+1} < s_{l+3} = s_l + 1 = 2(k+k'+1)$ the invariant follows. If Spoiler picks an event, $e_0^{x_m,s}$ say, in $\{e_0^{x_p,s_l}, e_0^{x_q,s_{l+2}}\}$ *before* the removed occurrence in σ then $m \in \{l, l+2\}$ and Preserver responds by $e_1^{x_m,s}$. Then $\mathit{sect}(e_0^{x_m,s}) = s = \mathit{sect}(e_1^{x_m,s}) \leq s_{l+2} = 2(k+k'+1)$. Similarly, if $e_0^{x_m,s}$ occurs *after* the removed occurrence then $m \in \{l+2, l+4\}$ and Preserver picks $e_1^{x_m,s+1}$. Then $\mathit{sect}(e_0^{x_m,s}) = s = \mathit{sect}(e_1^{x^m,s+1}) - 1 \geq 2(k+k') > 2(k-(k'+1))+1$ and in both cases the invariant is maintained. Finally, if Spoiler picks one of the remaining events $e_0^{x_i,s_i-1}$ with $i \geq l+5$ then Preserver responds with $e_1^{x_i,s_i}$. As $\mathit{sect}(e_0^{x_i,s_i-1}) = \mathit{sect}(e_1^{x_i,s_i}) - 1 \geq 2(k-(k'+1))+1$ the invariant is also maintained in this case.

Subcase II: $\mathit{sect}(e_0) = \mathit{sect}(e_1) - 1$. Here the futures of e_0 in T_0^k and e_1 in T_1^k both consist of $4k - \mathit{sect}(e_0)$ factors of $[abc]$ and are identical with respect to future moves. Hence Preserver can just "copy" the move made by Spoiler. with the obvious correspondence. □

6 The Expressiveness of TLC*

Let (Σ, I) be any trace alphabet with $\{a,b,c\} \subseteq \Sigma$ such that $a\ D\ c$ and $c\ D\ b$ but $a\ I\ b$. Consider $L = [abcabc]^* \subseteq TR(\Sigma, I)$ from the previous section.

Lemma 6.1 *L is definable in* $\mathrm{TLC}^*(\Sigma, I)$.

Proof: Our proof will in fact show that the future fragment of TLC with only *one* future chain quantifier of TLC* suffices to express L. First define

$$\begin{aligned}\alpha_{[abc]^*} \;=\; & AG(\bigwedge_{d\in\Sigma-\{a,b,c\}} {\sim}p_d) \wedge EX(p_a \wedge EX(p_c)) \wedge EX(p_b \wedge EX(p_c)) \wedge \\ & AG(p_c \wedge EX(\mathrm{tt}) \rightarrow EX(p_a \wedge EX(p_c)) \wedge EX(p_b \wedge EX(p_c)).\end{aligned}$$

It is easy to see that $T \models \alpha_{[abc]^*}$ iff $T \in [abc]^*$. We will then use existence of "zig-zagging" future causal chains to restrict to $[abcabc]^* \subset [abc]^*$ below. Define the future chain formula $\phi_{(acbc)^*}$ as follows.

$$\phi_{(acbc)^*} = p_a \wedge G(p_a \rightarrow X(p_c \wedge X(p_b \wedge X(p_c \wedge ({\sim}X\mathrm{tt} \vee Xp_a))))).$$

It's easy to see that $T, e \models E(\phi_{(acbc)^*})$ iff there exists a future causal chain ρ rooted at e such that $\lambda(\rho) \in (acbc)^* \subseteq \Sigma^*$. The statement of the lemma now follows by taking $\alpha_L = \alpha_{[abc]^*} \wedge ({\sim}EX\mathrm{tt} \vee EX(E(\phi_{(acbc)^*})))$. □

Putting all the pieces together, we can now state and prove the main result of the paper.

Theorem 6.2 *Let (Σ, I) be any trace alphabet. Then*

1. $\mathrm{TLC}(\Sigma, I) = \mathrm{TLC}^*(\Sigma, I)$ *if D is transitive.*
2. $\mathrm{TLC}(\Sigma, I) \subset \mathrm{TLC}^*(\Sigma, I)$ *if D is not transitive.*

Proof: Obviously $\mathrm{TLC}(\Sigma, I) \subseteq \mathrm{TLC}^*(\Sigma, I)$, so (2) follows easily from Lemma 5.1 and Lemma 6.1 as $(a, c), (c, b) \in D$ but $(a, b) \notin D$ witness that D is not transitive. Hence it suffices to prove (1).

Let (Σ, I) be a trace alphabet with D transitive, i.e. the graph (Σ, D) is a disjoint union of cliques $\{C_i\}_{i=1}^n$. Thus any trace $T \in TR(\Sigma, I)$ consists of disjoint C_i-labelled causal chains only initially connected by $\perp$. We can then define three mutually inductive translations $||\cdot||_{ev}$, $||\cdot||^+_{ch}$ and $||\cdot||^-_{ch}$ converting event formulas, future chain formulas and past chain formulas, respectively, of $\mathrm{TLC}^*(\Sigma, I)$ to formulas of $\mathrm{TLC}(\Sigma, I)$ as follows.

- $||p_a||_{ev} = p_a$ and the boolean connectives are as expected.
- $||\mathrm{co}(\alpha)||_{ev} = \mathrm{co}(||\alpha||_{ev})$.

- $||E(\phi)||_{ev} = ||\phi||^{+}_{ch}$ and $||E^{-}(\phi)||_{ev} = ||\phi||^{-}_{ch}$.
- $||\alpha||^{+}_{ch} = ||\alpha||^{-}_{ch} = ||\alpha||_{ev}$ and the boolean connectives are as expected.
- $||X\phi||^{+}_{ch} = EX(||\phi||^{+}_{ch})$ and $||X^{-}\phi||^{-}_{ch} = EX^{-}(||\phi||^{-}_{ch})$.
- $||\phi U\psi||^{+}_{ch} = EU(||\phi||^{+}_{ch}, ||\psi||^{+}_{ch})$ and $||\phi U^{-}\psi||^{-}_{ch} = EU(||\phi||^{-}_{ch}, ||\psi||^{-}_{ch})$.

By nested inductions one can show that for each $\alpha \in \mathrm{TLC}^{*}(\Sigma, I)$, $T, e \models \alpha$ iff $T, e \models ||\alpha||_{ev}$. As $||\alpha||_{ev} \in \mathrm{TLC}(\Sigma, I)$ the required conclusion follows. □

One can show that $\mathrm{TLC}^{*}(\Sigma, I)$ can be expressed within $\mathrm{MSO}(\Sigma, I)$. Moreover, it's not hard to see that $\mathrm{FO}(\Sigma, I)$, $\mathrm{TLC}(\Sigma, I)$ and $\mathrm{TLC}^{*}(\Sigma, I)$ are all expressively equivalent in the sequential case where $I = \emptyset$. Hence, we obtain the following corollary.

Corollary 6.3

1. $\mathrm{TLC}^{*}(\Sigma, \emptyset) \subset \mathrm{MSO}(\Sigma, \emptyset)$.
2. $\mathrm{TLC}^{*}(\Sigma, I) \not\subseteq \mathrm{FO}(\Sigma, I)$ *if D is not transitive.*

Proof: The proof of (1) follows from $\mathrm{TLC}^{*}(\Sigma, \emptyset) = \mathrm{FO}(\Sigma, \emptyset)$ and the well-known classical fact that $\mathrm{FO}(\Sigma, \emptyset) \subset \mathrm{MSO}(\Sigma, \emptyset)$.

For the proof of (2), let (Σ, I) be given and suppose D is not transitive. As before, we can then define $L = [abcabc]^{*}$. By a standard argument one can show that $((abc + bac)(abc + bac))^{*}$, the set of linearizations of L, is not a first-order definable language of strings over Σ. From [6] it then follows that L is not definable in $\mathrm{FO}(\Sigma, I)$. Lemma 6.1 now yields the required conclusion. □

We conclude by summarizing what is currently known about the relative expressive powers of the various linear time temporal logics over traces. A quick overview is displayed in Figure 2. A dotted (solid) arrow from A to B indicates that B is at least as expressive as (strictly more expressive than) A. Squiggled lines denote that the logics are incomparable to each other, while cut lines denote negation.

Theorem 6.2 and Corollary 6.3 have been applied to place TLC* in Figure 2. Like most other logics, its relationship to the first-order logic of traces needs to be pinned down exactly. Also, one might try to capture the expressive power of TLC* as (fragments of) either $\mathrm{FO}^{2}(\mathrm{TC})$, two-variable first-order

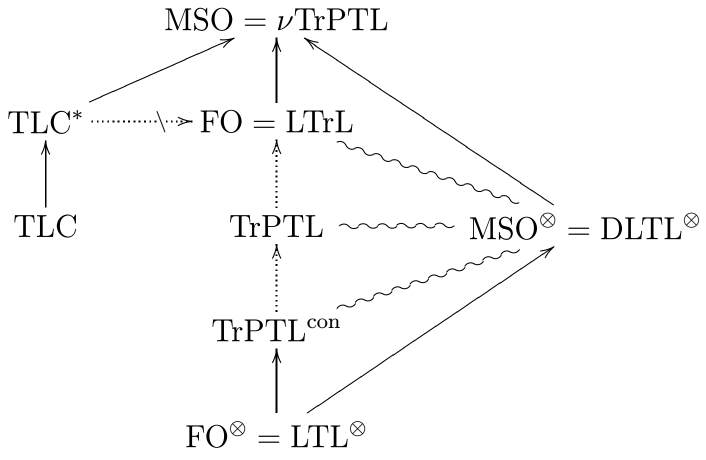

Figure 2: Overview of relative expressive powers

logic of traces augmented with the transitive-closure operator, or MSO with set quantifications restricted to causal chains. This constitutes an interesting open problem to be investigated in future work.

References

[1] Alur, R., Peled, D., Penczek, W.: Model checking of causality properties. Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science (LICS'95), IEEE Computer Society Press (1995) 90–100

[2] Bell Labs Design Automation: FormalCheck$^{\text{tm}}$. Further information can be obtained at `http://www.bell-labs.com/formalcheck/`

[3] Clarke, E. M., Emerson, E. A., Sistla, A. P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems **8**(2) (1986) 244–263

[4] Diekert, V., Gastin, P.: An expressively complete temporal logic without past tense operators for Mazurkiewicz traces. Proceedings of the 13th Annual Conference of the European Association for Computer Science Logic (CSL'99), Lecture Notes in Computer Science **1683**, Springer-Verlag (1999)

[5] Diekert, V., Rozenberg, G. (eds.): The book of traces. World Scientific (1995)

[6] Ebinger, W., Muscholl, A.: Logical definability on infinite traces. Theoretical Computer Science **154**(1) (1996) 67–84

[7] Etessami, K., Wilke, Th.: An until hierarchy for temporal logic. Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press (1996) 108–117

[8] Godefroid, P.: Partial-order methods for the verification of concurrent systems. Lecture Notes in Computer Science **1032**, Springer-Verlag (1996)

[9] Holzmann, G. J.: An overview of the SPIN model checker. In "On-the-fly Model Checking Tutorial", BRICS Autumn School on Verification, Note NS-96-6, BRICS, Department of Computer Science, University of Aarhus (1996)

[10] Mazurkiewicz, A.: Concurrent program schemes and their interpretations. Technical report PB-78, Department of Computer Science, University of Aarhus, Denmark (1977)

[11] Niebert, P.: A temporal logic for the specification and validation of distributed behaviour. Ph.D. thesis, University of Hildesheim (1997)

[12] Peled, D.: Partial order reduction: model checking using representatives. Proceedings of the 21st International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science **1113**, Springer-Verlag (1996) 93–112

[13] Pnueli, A.: The temporal logic of programs. Proceedings of the 18th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press (1977) 46–57

[14] Ramanujam, R.: Locally linear time temporal logic. Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press (1996) 118–127

[15] Thiagarajan, P. S.: A trace based extension of linear time temporal logic. Proceedings of the 9th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press (1994) 438–447

[16] Thiagarajan, P. S., Henriksen, J. G.: Distributed versions of linear time temporal logic: A trace perspective. In Reisig and Rozenberg (Eds.), Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science **1491**, Springer-Verlag (1998) 643–681

[17] Thiagarajan, P. S., Walukiewicz, I.: An expressively complete linear time temporal logic for Mazurkiewicz traces. Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press (1997) 183–194

[18] Valmari, A.: A stubborn attack on state explosion. Formal Methods in Systems Design **1** (1992) 285–313

[19] Walukiewicz, I.: Difficult configurations — on the complexity of LTrL (extended abstract). Proceedings of the 25th International Colloquium on Automata, Languages and Programming. Lecture Notes in Computer Science **1443**, Springer-Verlag (1998) 140–151

Recent BRICS Report Series Publications

RS-99-26 **Jesper G. Henriksen.** ***An Expressive Extension of TLC.*** **September 1999. 20 pp. To appear in Thiagarajan and Yap, editors,** ***Fifth Asian Computing Science Conference,*** **ASIAN '99 Proceedings, LNCS, 1999.**

RS-99-25 **Gerth Stølting Brodal and Christian N. S. Pedersen.** ***Finding Maximal Quasiperiodicities in Strings.*** **September 1999. 20 pp.**

RS-99-24 **Luca Aceto, Willem Jan Fokkink, and Chris Verhoef.** ***Conservative Extension in Structural Operational Semantics.*** **September 1999. 23 pp. To appear in the** ***Bulletin of the EATCS.***

RS-99-23 **Olivier Danvy, Belmina Dzafic, and Frank Pfenning.** ***On proving syntactic properties of CPS programs.*** **August 1999. 14 pp. To appear in Gordon and Pitts, editors,** ***3rd Workshop on Higher Order Operational Techniques in Semantics,*** **HOOTS '99 Proceedings, ENTCS, 1999.**

RS-99-22 **Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir.** ***On the Two-Variable Fragment of the Equational Theory of the Max-Sum Algebra of the Natural Numbers.*** **August 1999. 22 pp.**

RS-99-21 **Olivier Danvy.** ***An Extensional Characterization of Lambda-Lifting and Lambda-Dropping.*** **August 1999. 13 pp. Extended version of an article to appear in** ***Fourth Fuji International Symposium on Functional and Logic Programming,*** **FLOPS '99 Proceedings (Tsukuba, Japan, November 11–13, 1999). This report supersedes the earlier report BRICS RS-98-2.**

RS-99-20 **Ulrich Kohlenbach.** ***A Note on Spector's Quantifier-Free Rule of Extensionality.*** **August 1999. 5 pp. To appear in** ***Archive for Mathematical Logic.***

RS-99-19 **Marcin Jurdziński and Mogens Nielsen.** ***Hereditary History Preserving Bisimilarity is Undecidable.*** **June 1999. 18 pp.**

RS-99-18 **M. Oliver Möller and Harald Rueß.** ***Solving Bit-Vector Equations of Fixed and Non-Fixed Size.*** **June 1999. 18 pp. Revised version of an article appearing under the title** ***Solving Bit-Vector Equations*** **in Gopalakrishnan and Windley, editors,** ***Formal Methods in Computer-Aided Design: Second International Conference,*** **FMCAD '98 Proceedings, LNCS 1522, 1998, pages 36–48.**