

An Expressively Complete Linear Time Temporal Logic for Mazurkiewicz Traces¹

P.S. Thiagarajan	I. Walukiewicz ²
SPIC Mathematical Institute	Institute of Informatics
92 G.N. Chetty Road	Warsaw University
T. Nagar, Chennai-600 017	Banacha 2
India.	02-096 Warsaw, Poland

Abstract

A basic result concerning *LTL*, the propositional temporal logic of linear time, is that it is expressively complete; it is equal in expressive power to the first order theory of sequences. We present here a smooth extension of this result to the class of partial orders known as Mazurkiewicz traces. These partial orders arise in a variety of contexts in concurrency theory and they provide the conceptual basis for many of the partial order reduction methods that have been developed in connection with *LTL*-specifications.

We show that *LTrL*, our linear time temporal logic, is equal in expressive power to the first order theory of traces when interpreted over (finite and) *infinite* traces. This result fills a prominent gap in the existing logical theory of infinite traces. *LTrL* also constitutes a characterisation of the so called trace consistent (robust) *LTL*-specifications. These are specifications expressed as *LTL* formulas that do not distinguish between different linearisations of the same trace and hence are amenable to partial order reduction methods.

¹This work was mainly done at BRICS, Basic Research in Computer Science, Centre of the Danish National Research Foundation, Computer Science Department, Aarhus University, Denmark.

²The author was partially supported by Polish KBN grant No. 8 T11C 002 11.

1 Introduction

We propose a linear time temporal logic called *LTrL* whose models are Mazurkiewicz traces. From its inception [16] the class of labelled partial orders known as (Mazurkiewicz) traces has played a fundamental role in the theory of distributed systems. In particular, traces constitute the natural tool for capturing the non-interleaved linear time behaviour of models such as Petri nets, event structures and distributed transition systems of various kinds [30].

The theory of traces is well-developed [3]. This theory may be viewed as a smooth generalization of the classical theory of sequences. It turns out that most of the algebraic, automata-theoretic and logical results concerning sequences [27] have a natural extension to the setting of traces. There has been however one prominent gap to date in the logical theory of traces. Our main result concerning *LTrL* fills this gap.

To bring this out, we recall the famous theorem of Kamp [13] extended by Gabbay et.al. [8]. It says that *LTL*, the linear time temporal logic interpreted over sequences has the same expressive power as the first order theory of sequences. The surprising aspect of this result being the mismatch between the ability to define an infinite number of operators of increasing arities in the first order theory and the bounded number of operators (one binary and one unary operator) admitted by *LTL*. For the class of partial orders as a whole, it is known that there can be no such temporal logic [8]. So far, it has been an open problem to determine whether the class of traces viewed as a subclass of partial orders admits a temporal logic (with a bounded number of operators) which has the same expressive power as the first order theory of traces. Our main result solves this problem by providing a positive solution. In other words, we show that that *LTrL* with its bounded number of operators has the same expressive power as the first order theory of traces.

There is also a more pragmatic motivation for studying temporal logics interpreted over traces. To bring this out, we first recall that, as first suggested by Pnueli [23], *LTL* is often interpreted over the runs of a distributed system. It is known that these runs can be grouped together into equivalence classes; two runs are equated in case they differ only in the order

in which causally independent occurrences of events are recorded. In other words, each equivalence class corresponds to all possible linearizations of a single partially ordered stretch of behaviour. In many settings, the partial orders that arise in this fashion are traces. Further, it is also often the case that the property expressed by an *LTL* specification is insensitive to a choice of linearizations in the sense that either all members of an equivalence class satisfy the property or none do. Such properties are often called robust or trace consistent properties. For verifying that all the runs of a distributed system satisfies a trace consistent requirement, it suffices to check that the requirement is met by at least one member of each equivalence class of runs. The resulting savings in the computational resources used during the verification of trace consistent requirements can be substantial. This is the insight that underlies many of the so called partial order reduction techniques [11, 20, 31].

There is an alternative way to exploit the non-sequential nature of the behaviour of distributed systems and the consequent partial order based reduction techniques. It consists of developing temporal logics that can be directly interpreted over the partial orders corresponding to equivalence classes of runs. The formulas of such logics will describe only trace consistent properties. Hence the associated verification task will be amenable to partial order based reduction methods. This is, in retrospect, the key feature of the *branching time* temporal logic ISTL introduced by Katz and Peled [14]. The explicit connection between ISTL and traces was later formalized and exploited by Peled and Pnueli [21] to derive proof rules for reasoning about the partially ordered runs of a concurrent program.. At present we do not know of a characterization of the expressive power of ISTL and its variants.

In a linear time setting, there is an important criterion one could use to judge the expressive power of a temporal logic interpreted over traces. One could demand that such a logic should capture exactly the trace consistent properties that can be defined by *LTL* formulas. It seems difficult to use *LTL* itself to capture trace consistent properties, say, through syntactic restrictions. One reason could be that the problem of deciding whether the property described by an *LTL* formula is trace consistent happens to be PSPACE-complete [22].

From the work of Ebinger and Muscholl [6] it is not difficult to con-

clude the following: A linear time temporal logic interpreted over traces will capture exactly the *LTL*-definable trace consistent properties if and only if this logic is expressively equivalent to the first order theory of traces. Thus an important corollary of our main result is that *LTrL* captures exactly the *LTL*-definable trace consistent properties.

Starting with [25] a number of linear time temporal logics for traces have been proposed in the literature [1, 5, 18, 24]. None of these studies have been able to exhibit a logic patterned after *LTL* which is equivalent in expressive power to the first order theory of traces. These logics also have a semantics which has a strong “local” flavour. As a result they can not formulate in a natural way arbitrary global liveness and safety properties. In contrast, in *LTrL* one can transparently express global liveness and safety properties of all kinds.

There is however a price to be paid for this transparency. The second author of this paper has recently established a non-elementary time lower bound for the satisfiability problem for *LTrL* [29]. Hence it is clear that from a practical standpoint *LTrL* is not the final stop in the search for the “right” linear time temporal logic for traces. However, we feel that this logic represents a vital step forward towards achieving this goal. We also feel that the novel techniques developed to establish our main result will eventually lead to a suitable variant of *LTrL* which, while remaining expressively complete, will also admit a decision procedure with a more reasonable time complexity.

The only available expressiveness results for temporal logics over traces are due to Ebinger [5] and Niebert [19]. Ebinger’s logic, called *TLPO*, has both previous state and since modalities. These past modalities are extensively used in the attempt to prove that *TLPO* is expressively complete when interpreted over *finite* traces. This proof does not extend to infinite traces. In contrast, *LTrL* uses only a very restricted previous state modality. And it is expressively complete over the domain of infinite traces as well.

Niebert has recently formulated a fixed point temporal logic called $\nu TrPTL$ interpreted over infinite traces. This logic is shown to be equal in expressive power to the monadic second order theory of traces. Further, the satisfiability problem for $\nu TrPTL$ is shown to be solvable in essentially exponential time. A drawback of this interesting new development is that the formulas

of $\nu TrPTL$ are required to satisfy awkward syntactic restrictions. It is also not clear how easy it is to formulate global properties of interest in $\nu TrPTL$. Due to the fundamental role of fixed point operators it is also not possible to cut out a natural fragment of $\nu TrPTL$ which will capture exactly the LTL -definable trace consistent properties.

The outline of the paper is as follows. In the next section we introduce traces. The first order theory of traces as well as the syntax and semantics of $LTrL$ are presented in section 3. This leads to the formulation of the main result and its corollaries. The major ingredients of the proof of the main result are: a decomposition result for infinite traces, which is an easy version of the Feferman-Vaught theorem for disjoint sums [7] and a new normal form linearisation of traces. It is the use of the Feferman-Vaught result and the new normal form that takes us past the key technical hurdles. The proof is presented in sections 4–9. A proof outline of the difficult half of the main result is given in section 5.

2 Traces

A (Mazurkiewicz) *trace alphabet* is a pair (Σ, I) where Σ is a finite set of actions and $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric *independence relation*. $D = (\Sigma \times \Sigma) - I$ is called the dependency relation. Through the rest of the paper we fix a trace alphabet (Σ, I) and we will often refer to it implicitly. We let a, b range over Σ .

We shall view (Mazurkiewicz) trace as a restricted Σ -labelled poset. Let (E, \leq, λ) be a Σ -labelled poset. In other words, (E, \leq) is a poset and $\lambda : E \rightarrow \Sigma$ is a labelling function. For $Y \subseteq E$ we define $\downarrow Y = \{x \mid \exists y \in Y. x \leq y\}$ and $\uparrow Y = \{x \mid \exists y \in Y. y \leq x\}$. In case $Y = \{y\}$ is a singleton we shall write $\downarrow y$ ($\uparrow y$) instead of $\downarrow \{y\}$ ($\uparrow \{y\}$). We also let \triangleleft be the relation: $x \triangleleft y$ iff $x < y$ and $\forall z \in E. x \leq z \leq y$ implies $x = z$ or $z = y$.

A *trace* (over (Σ, I)) is a Σ -labelled poset $T = (E, \leq, \lambda)$ satisfying:

- (T1) $\forall e \in E. \downarrow e$ is a finite set
- (T2) $\forall e, e' \in E. e \triangleleft e' \Rightarrow \lambda(e) D \lambda(e')$.
- (T3) $\forall e, e' \in E. \lambda(e) D \lambda(e') \Rightarrow e \leq e'$ or $e' \leq e$.

We shall refer to members of E as *events*. The trace $T = (E, \leq, \lambda)$ is said to be *finite* if E is a finite set. Otherwise it is an infinite trace. Note that E

is always a countable set. T is said to be *non-empty* in case $E \neq \emptyset$. We let $TR^{fin}(\Sigma, I)$ be the set of finite traces and $TR^{inf}(\Sigma, I)$ be the set of infinite traces over (Σ, I) and set $TR(\Sigma, I) = TR^{fin}(\Sigma, I) \cup TR^{inf}(\Sigma, I)$. Often we will write TR^{fin} instead of $TR^{fin}(\Sigma, I)$ etc.

Let $T = (E, \leq, \lambda)$ be a trace. The finite prefixes of T , to be called configurations, will play a crucial role in what follows. A configuration of T is a finite subset $c \subseteq E$ such that $c = \downarrow c$. We let C_T be the set of configurations of T and let c, c', c'' range over C_T . Note that \emptyset , the empty set, is a configuration and $\downarrow e$ is a configuration for every $e \in E$. Finally, the transition relation $\rightarrow_T \subseteq C_T \times \Sigma \times C_T$ is given by: $c \xrightarrow{a}_T c'$ iff there exists $e \in E$ such that $\lambda(e) = a$ and $e \notin c$ and $c' = c \cup \{e\}$. It is easy to see that if $c \xrightarrow{a}_T c'$ and $c \xrightarrow{a}_T c''$ then $c' = c''$. The configurations of a trace serve as the finite stages of the distributed run modeled by the trace. The formulas of our temporal logic will be interpreted at the configurations of a trace.

3 The Main Result

The first order theory of traces is formulated by assuming a countable set of individual variables $Var = \{x, y, z, \dots\}$; a family of unary predicates $\{R_a\}_{a \in \Sigma}$; and a binary predicate \leq . Then $FO(\Sigma, I)$, the set of formulas in the first order theory of traces (over (Σ, I)), is given by the syntax:

$$FO(\Sigma, I) ::= R_a(x) \mid x \leq y \mid \sim \varphi \mid \varphi \vee \varphi' \mid (\exists x)\varphi$$

Thus the syntax does not explicitly involve I . However, it is reflected in \leq that will be interpreted as the partial order relation associated with a trace which does indeed respect the independence relation I .

Given a trace $T = (E, \leq, \lambda)$ and an associated valuation $V : Var \rightarrow E$, the relation $T \models_V^{FO} \varphi$ will denote that T is a model of $\varphi \in FO(\Sigma, I)$ under the valuation V . This notion is defined in the expected manner. In particular, $T \models_V^{FO} R_a(x)$ iff $\lambda(V(x)) = a$ and $T \models_V^{FO} x \leq y$ iff $V(x) \leq V(y)$. As usual, a sentence is a formula with no free variables. L_φ will denote the set of models of the sentence φ . More precisely,

$$L_\varphi = \{T : T \in TR \text{ and } T \models^{FO} \varphi\}$$

We will say that $L \subseteq TR$ is *FO-definable* iff there exists a sentence $\varphi \in FO(\Sigma, I)$ such that $L = L_\varphi$.

The set of formulas of our linear time temporal logic of traces ($LTrL$) is defined as follows:

$$LTrL(\Sigma, I) ::= \underline{tt} \mid \sim \alpha \mid \alpha \vee \beta \mid \langle a \rangle \alpha \mid \alpha U \beta \mid \langle a^{-1} \rangle \underline{tt}$$

Thus the next state modality is indexed by actions. There is also a very restricted version of the previous state modality. Indeed the number of past formulas is bounded by the size of Σ . For achieving the present aims, there is no need for atomic propositions. It is worth mentioning that if atomic propositions are to be introduced then the valuations must be required to respect the independence relation in a suitable fashion. The logic will become undecidable otherwise. More on this issue can be found in the concluding section. In the current framework, a model of $LTrL$ is just a trace $T = (E, \leq, \lambda)$. The relation $T, c \models \alpha$ will denote that $\alpha \in LTrL(\Sigma, I)$ is satisfied at the configuration $c \in C_T$. It is defined via:

- $T, c \models \underline{tt}$. Furthermore \sim and \vee are interpreted in the usual way.
- $T, c \models \langle a \rangle \alpha$ iff $\exists c' \in C_T. c \xrightarrow{a}_T c'$ and $T, c' \models \alpha$.
- $T, c \models \alpha U \beta$ iff $\exists c' \in C_T. c \subseteq c'$ and $T, c' \models \beta$ and $\forall c'' \in C_T. c \subseteq c'' \subset c'$ implies $T, c'' \models \alpha$.
- $T, c \models \langle a^{-1} \rangle \underline{tt}$ iff $\exists c' \in C_T. c' \xrightarrow{a}_T c$.

The derived “sometime” and “always” modalities have expected semantics. More precisely, with $\diamond \alpha \stackrel{\Delta}{\Leftrightarrow} \underline{tt} U \alpha$ and $\square \alpha \stackrel{\Delta}{\Leftrightarrow} \sim \diamond \sim \alpha$, we have: $T, c \models \square \alpha$ iff $\forall c' \in C_T. c \subseteq c'$ implies $T, c' \models \alpha$. Thus arbitrary liveness and safety properties interpreted over the global states of a distributed system can be formulated in $LTrL$. A variety of models of distributed systems are available in the literature whose non-interleaved linear time semantics can be given in terms of traces. Elementary net systems, 1-safe Petri nets, trace transition systems [30] and networks of finite state automata that communicate by synchronizing on common actions [17] are typical examples. For all these system models, one can use $LTrL$ formulas to specify the properties that must be satisfied by all trace runs. An example of this approach is given in Section 11.

With each formula $\alpha \in LTrL(\Sigma, I)$, we can associate a set of traces as follows:

$$L_\alpha = \{T \in TR \mid T, \emptyset \models \alpha\}.$$

We say that $L \subseteq TR$ is *LTrL-definable* iff there exists a formula $\alpha \in LTrL(\Sigma, I)$ such that $L = L_\alpha$.

Our main result can now be stated.

Theorem 1

Let $L \subseteq TR^{inf}$. Then L is FO-definable iff L is LTrL-definable.

Indeed this result goes through in case $L \subseteq TR^{fin}$ or $L \subseteq TR$. We note that in case $I = \emptyset$, Theorem 1 is just the expressiveness result of [8] in a different and slightly weakened (because of the past modalities) form.

The theorem is proved by showing separately the two implications:

Lemma 2 Let $\alpha \in LTrL(\Sigma, I)$. Then there exists $\varphi \in FO(\Sigma, I)$ such that for every $T \in TR^{inf}$: $T, \emptyset \models \alpha$ iff $T \models^{FO} \varphi$.

Lemma 3 Let $\varphi \in FO(\Sigma, I)$. Then there exists $\alpha \in LTrL(\Sigma, I)$ such that for every $T \in TR^{inf}$: $T \models^{FO} \varphi$ iff $T, \emptyset \models \alpha$.

The proof of the first lemma is straightforward. It consists of coding the semantics of $LTrL(\Sigma, I)$ into $FO(\Sigma, I)$. The proof of the second lemma is much more involved. We will show it first for finite and restricted infinite traces called perpetual directed traces. Then we will use some decomposition and composition results to show the lemma for all traces.

Let us finish this section with some corollaries of our main result. As the first order theory of traces is decidable [6] and the translation given in the proof of Lemma 2 is constructive we immediately obtain:

Corollary 4 The satisfiability problem for *LTrL* is decidable.

To bring out one more consequence of Theorem 1, we shall define *LTL*(Σ), linear time temporal logic interpreted over Σ -sequences. We will use Σ^* and Σ^ω to denote the set of finite and infinite sequences over Σ respectively. We will use Σ^∞ for $\Sigma^* \cup \Sigma^\omega$.

The syntax of *LTL*(Σ) is given by:

$$LTL(\Sigma) ::= \underline{tt} \mid \sim \hat{\alpha} \mid \hat{\alpha} \vee \hat{\beta} \mid \langle a \rangle \hat{\alpha} \mid \hat{\alpha} \mathcal{U} \hat{\beta}.$$

For $\sigma \in \Sigma^\infty$, let $\text{prf}(\sigma)$ denote the set of finite prefixes of σ and let $\tau \sqsubseteq \tau'$ denote that τ is a prefix of τ' . Then $\sigma, \tau \models \hat{\alpha}$ will stand for $\hat{\alpha}$ being satisfied at the prefix τ of σ . This notion is defined in the usual way.

- $\sigma, \tau \models \underline{\text{tt}}$. The connectives \sim and \vee are interpreted in the standard fashion.
- $\sigma, \tau \models \langle a \rangle \hat{\alpha}$ iff $\tau a \in \text{prf}(\sigma)$ and $\sigma, \tau a \models \hat{\alpha}$.
- $\sigma, \tau \models \hat{\alpha} \mathcal{U} \hat{\beta}$ iff $\exists \tau' \in \text{prf}(\sigma)$ such that $\tau \sqsubseteq \tau'$ and $\sigma, \tau' \models \hat{\beta}$. Moreover for every $\tau'' \in \text{prf}(\sigma)$, if $\tau \sqsubseteq \tau'' \sqsubset \tau'$ then $\sigma, \tau'' \models \hat{\alpha}$.

Next, let $T = (E, \leq, \lambda) \in TR$. Then $\sigma \in \Sigma^\infty$ is a *linearisation* of T iff there exists a map $\rho : \text{prf}(\sigma) \rightarrow C_T$, such that, the following conditions are met:

- (i) $\rho(\epsilon) = \emptyset$ (ϵ is the null string)
- (ii) $\forall \tau a \in \text{prf}(\sigma)$ with $\tau \in \Sigma^*$, $\rho(\tau) \xrightarrow{a}_T \rho(\tau a)$
- (iii) $\forall e \in E \exists \tau \in \text{prf}(\sigma)$. $e \in \rho(\tau)$.

The function ρ will be called a *run map* of the linearisation σ . Note that the run map of a linearisation is unique. In what follows, we shall let $\text{lin}(T)$ to be the set of linearisations of the trace T . The notion of linearisation induces the standard equivalence relation $\approx_I \subseteq \Sigma^\infty \times \Sigma^\infty$ via: $\sigma \approx_I \sigma'$ iff there exists a trace T , such that, $\sigma, \sigma' \in \text{lin}(T)$. It is well-known that there is a natural correspondence between $\Sigma^\infty / \approx_I$ and $TR^\infty(\Sigma, I)$. A formula $\hat{\alpha}$ is said to be *trace consistent* if for every $\sigma, \sigma' \in \Sigma^\infty$ such that $\sigma \approx_I \sigma'$, whenever $\sigma, \varepsilon \models \hat{\alpha}$ then $\sigma', \varepsilon \models \hat{\alpha}$.

As mentioned earlier, specifications that are formulated as trace consistent formulas can be often verified efficiently using partial order reduction techniques. *LTrL* provides a characterisation of trace consistent *LTL* formulas in the following sense.

- Corollary 5**
- (i) For every formula $\alpha \in LTrL(\Sigma, I)$ there is a trace consistent formula $\hat{\alpha} \in LTL(\Sigma)$, such that $\bigcup \{ \text{lin}(T) : T, \emptyset \models \alpha \} = \{ \sigma : \sigma, \varepsilon \models \hat{\alpha} \}$.
 - (ii) For every trace consistent *LTL*(Σ) formula $\hat{\alpha}$ there is a *LTrL*(Σ, I) formula α such that $\bigcup \{ \text{lin}(T) : T, \emptyset \models \alpha \} = \{ \sigma : \sigma, \varepsilon \models \hat{\alpha} \}$.

4 From $LTrL(\Sigma, I)$ to $FO(\Sigma, I)$

In this section we will show Lemma 2 which is an easy part of Theorem 1

Proof (of Lemma 2)

In $FO(\Sigma, I)$ the variables range over events, but we can use a finite set of variables to represent a configuration. Intuitively a set of variables X represents in a given valuation $V : Var \rightarrow E$ the configuration $c_V^X = \{e : \exists z \in X. e \leq V(z)\}$.

For every set of variables X and every formula α of $LTrL$ we will construct a formula φ_α^X of $FO(\Sigma, I)$ with free variables in the set X . This formula will have the property that for every valuation $V : Var \rightarrow E$:

$$T \models_V^{FO} \varphi_\alpha^X \quad \text{iff} \quad T, c_V^X \models \alpha \quad (1)$$

In particular taking $X = \emptyset$ we will obtain the thesis of the lemma.

The construction proceeds by structural induction on α . If $\alpha = \underline{tt}$ then for every X we put $\varphi_\alpha^X = \forall z. (z \leq z)$. The cases for disjunction and negation are also obvious.

Suppose $\alpha = \langle a \rangle \beta$. Let $X = \{x_1, \dots, x_k\}$ (this set may be empty). We let φ_α^X to be:

$$\exists y. R_a(y) \wedge \varphi_\beta^{X \cup \{y\}} \wedge \left(\bigwedge_{i=1, \dots, k} y \not\leq x_i \right) \wedge \left(\forall z. z < y \Rightarrow \bigvee_{i=1, \dots, k} z \leq x_i \right)$$

Suppose $\alpha = \beta U \gamma$. First, for two sets of variables Y, Z we define the formulas

$$\text{Below}(Y, Z) = \bigwedge_{y \in Y} \left(\bigvee_{z \in Z} y \leq z \right)$$

$$\text{SBelow}(Y, Z) = \text{Below}(Y, Z) \wedge \neg \text{Below}(Z, Y)$$

Intuitively formula $\text{Below}(X, Y)$ says that all the events in the configuration represented by Y belong to a configuration represented by Z . The formula $\text{SBelow}(X, Y)$ says the same plus the fact that the configurations are not equal. With the help of this formula we define φ_α^X for $X \neq \emptyset$ by:

$$\begin{aligned} \exists Z. \text{Below}(X, Z) \wedge \varphi_\gamma^Z \wedge \\ \forall Y. (\text{Below}(X, Y) \wedge \text{SBelow}(Y, Z)) \Rightarrow \varphi_\beta^Y \end{aligned}$$

In the above the quantifier $\exists Z$ is an abbreviation of $\exists z_1, \dots, \exists z_{|\Sigma|}$. Similarly for $\forall Y$. We let φ_α^\emptyset to be:

$$\varphi_\gamma^\emptyset \vee \exists Z. \varphi_\gamma^Z \wedge \varphi_\beta^\emptyset \wedge \forall Y. \text{SBelow}(Y, Z) \Rightarrow \varphi_\beta^Y$$

Finally, if $\alpha = \langle a^{-1} \rangle_{\text{tt}}$ then the formula φ_α^X is

$$\bigvee_{x \in X} \left(R_a(x) \wedge \bigwedge_{x' \in X} x \neq x' \Rightarrow x \not\leq x' \right)$$

Here $x \neq x'$ is an abbreviation for $x \not\leq x' \wedge x' \not\leq x$.

By structural induction on α one can show that the condition (1) is satisfied.

□

5 From $FO(\Sigma, I)$ to $LTrL(\Sigma, I)$ for definable linearizations

The goal of this section is to give an outline of the proof of Lemma 3. First we will assume that we have some notion of a canonical linearization of traces. Further we will assume that there is an $LTrL$ formula which can, in some sense, describe this canonical linearization. With these assumptions in place, we will outline a proof of Lemma 3. Unfortunately there can be no such notion of a linearization that can be captured by a fixed $LTrL$ formula and which works for all traces. Towards the end of the section we will explain how we get around this problem by restricting our attention to special kinds of traces.

The first order theory of Σ -words will be denoted $FO(\Sigma)$ and it will have the same syntax as $FO(\Sigma, I)$. The only difference is that the binary relation \leq is to be viewed as the linear order relation over the positions of a Σ -word. Viewed differently, $FO(\Sigma)$ is just $FO(\Sigma, I)$ with $I = \emptyset$. The basic details concerning $FO(\Sigma)$ can be found in [27].

We will not reprove here the famous equivalence in expressive power between $FO(\Sigma)$ and $LTL(\Sigma)$ [13]. Rather, we will use this result and work with trace consistent fragment of $LTL(\Sigma)$ instead of $FO(\Sigma, I)$. This is possible by the following lemma which was observed in a slightly different setting in [6].

Lemma 6 Let φ be a sentence in $FO(\Sigma, I)$. Then there exists a *trace consistent* $\hat{\alpha} \in LTL(\Sigma)$ such that for every $T \in TR$: $T \models^{FO} \varphi$ iff $\sigma, \varepsilon \models \hat{\alpha}$ for *some* linearisation σ of T .

Proof

In what follows, the semantic relation of satisfiability associated with the sentences of $FO(\Sigma)$ will be denoted \models^{fo} . A simple but basic observation essentially due to Wolfgang Thomas [28] can be stated as:

Observation 6.1 For every sentence $\varphi \in FO(\Sigma, I)$ there exists a sentence $\hat{\varphi} \in FO(\Sigma)$ such that for every trace T : $T \models^{FO} \varphi$ iff $u \models^{fo} \hat{\varphi}$ for every $u \in lin(T)$.

Recall that $lin(T)$ is the set of linearisations of T . Now, let $T = (E, \leq, \lambda)$ be a trace, let $u \in lin(T)$ and let $\rho : \text{prf}(u) \rightarrow C_T$ the associated run map. Suppose that $e \in E$ and $\lambda(e) = a$. Then there exists a unique $\tau a \in \text{prf}(u)$ such that $e \notin \rho(\tau)$ and $e \in \rho(\tau a)$. Let us call this τa the occurrence of e in u . It is not difficult to show that $e < e'$ in T with $e, e' \in E$ iff there exists $\tau_0 a_0, \tau_1 a_1, \dots, \tau_n a_n \in \text{prf}(u)$ such that the following conditions are satisfied.

- $\tau_0 a_0$ is the occurrence of e and $\tau_n a_n$ is the occurrence e' in u .
- $\tau_0 a_0 \sqsubseteq \tau_1 a_1 \sqsubseteq \dots \sqsubseteq \tau_n a_n$.
- $1 \leq n \leq |\Sigma|$ and $a_i D a_{i+1}$ for $0 \leq i < n$.

All these conditions can be expressed in $FO(\Sigma)$ and this easily leads to Observation 6.1.

Now, by the expressiveness results of [8, 32], for each sentence $\hat{\varphi} \in FO(\Sigma)$ there exists $\hat{\alpha} \in LTL(\Sigma)$ such that:

$$\{u \in \Sigma^\infty \mid u \models^{fo} \hat{\varphi}\} = \{u \in \Sigma^\infty \mid u, \varepsilon \models \hat{\alpha}\}.$$

The lemma now follows at once from the definition of trace consistent formulas. □

Let us now show how to translate trace consistent $LTL(\Sigma)$ formulas to $LTrL(\Sigma, I)$ formulas assuming that we have a formula defining canonical linearisations of traces.

Definition 7 (i) A map $nrl : TR(\Sigma, I) \rightarrow \Sigma^\infty$ is called a linearization map iff $nrl(T) \in lin(T)$ for every trace T . Abstractly, nrl picks out one particular “normal” linearization of each trace.

(ii) A formula, say, $NRC \in LTrL(\Sigma, I)$ is said to *capture* the linearization map nrl iff the following condition holds for every trace T .

- Suppose $nrl(T) = \sigma$ and $\rho : prof(\sigma) \rightarrow C_T$ is the run map of σ . Then $\forall c \in C_T. T, c \models NRC$ iff there exists $\tau \in prof(\sigma)$ such that $\rho(\tau) = c$.

Thus the magic formula NRC (if it exists) is such that it holds exactly at the configurations in the image of the run map associated with the normal linearization.

Now assume that $nrl : TR(\Sigma, I) \rightarrow \Sigma^\infty$ is a linearization map and NRC is a formula in $LTrL(\Sigma, I)$ which captures nrl .

Using NRC , we now define the map $\|\cdot\| : LTL(\Sigma) \rightarrow LTrL(\Sigma, I)$ via :

$$\begin{aligned} \|\underline{tt}\| &= \underline{tt} & \|\sim \hat{\alpha}\| &= \sim \|\hat{\alpha}\| & \|\hat{\alpha} \vee \hat{\beta}\| &= \|\hat{\alpha}\| \vee \|\hat{\beta}\| \\ \|\langle a \rangle \hat{\alpha}\| &= \langle a \rangle (NRC \wedge \|\hat{\alpha}\|) \\ \|\hat{\alpha} \mathcal{U} \hat{\beta}\| &= (NRC \supset \|\hat{\alpha}\|) \mathcal{U} (NRC \wedge \|\hat{\beta}\|) \end{aligned}$$

By an easy structural induction one can show:

Lemma 8 Let $\hat{\alpha}$ be a formula of $LTL(\Sigma)$. For every trace T and its linearisation σ defined by nrl (i.e. $nrl(T) = \sigma$) we have: $\sigma, \varepsilon \models \hat{\alpha}$ iff $T, \emptyset \models \|\hat{\alpha}\|$.

Hence to establish Lemma 3 it would suffice to fix a linearization map and a formula in $LTrL$ which captures it. Unfortunately such a formula can not exist for any chosen linearization map (except of course in the degenerate case where $I = \emptyset$). This is because it does not even exist for a trace alphabet consisting of just two letters which are independent. An infinite trace over such an alphabet consists of two disconnected infinite sequences. Courcelle [2] has shown that there is no monadic second order formula defining a linearization of this trace.

Our approach to proving Lemma 3 is the following. We identify a restricted type of an infinite trace called a perpetual directed trace. We show that every (infinite) trace can be canonically decomposed into an initial finite

trace followed by a bounded number of perpetual directed traces. We then show how to accomplish for perpetual directed traces what was not possible for all traces. Namely, we define a *partial* linearization map defined only on finite and perpetual directed traces and a formula *NRC* which captures this map whenever it is applied to a finite or a perpetual directed trace. To extend Lemma 3 to all traces we first show how to decompose a formula in $FO(\Sigma, I)$ into a bounded number of formulas that talk about the finite initial part and the perpetual directed parts of a trace. After translating such decomposed formulas into *LTrL* via Lemma 3, we show how the resulting *LTrL* formulas can be combined together to talk about a whole trace.

6 Decomposition of traces

Let $T = (E, \leq, \lambda)$ be a trace. The alphabet of T is denoted $alph(T)$ and is given by: $alph(T) = \{\lambda(e) \mid e \in E\}$. We define as $\Sigma_T^{fin} = \{a : \lambda^{-1}(a) : \text{is a finite set}\}$.

Definition 9 (Perpetual and directed traces) (i) A trace T is called *perpetual* if it is non-empty and $\Sigma_T^{fin} = \emptyset$. Hence every perpetual trace is infinite but converse is not always true.

(ii) The trace T is called *directed* iff every two events in E have an upper bound, i.e., for every $e_1, e_2 \in E$ there exists e , such that, $e_1 \leq e$ and $e_2 \leq e$.

We now define the Σ -labelled posets $fin(T)$ and $inf(T)$ via:

$$fin(T) = (E_{fin}, \leq_{fin}, \lambda_{fin}) \text{ and } inf(T) = (E_{inf}, \leq_{inf}, \lambda_{inf})$$

where $E_{fin} = \{e : \exists e'. e \leq e' \text{ and } \lambda(e') \in \Sigma_T^{fin}\}$ and $E_{inf} = E - E_{fin}$. Furthermore, \leq_{fin} (\leq_{inf}) is \leq restricted to $E_{fin} \times E_{fin}$ ($E_{inf} \times E_{inf}$) and λ_{fin} (λ_{inf}) is λ restricted to E_{fin} (E_{inf}). The following observation is an easy consequence of the definitions.

Proposition 10 For every trace T , $fin(T)$ is a finite trace. Further, $inf(T)$ is a perpetual trace iff T is an infinite trace.

Next we decompose $inf(T)$.

Proposition 11 Let $T = (E, \leq, \lambda)$ be a perpetual trace. Then there exists a unique family of traces $\{T_i = (E_i, \leq_i, \lambda_i)\}_{i=1}^m$ with $m \leq |\Sigma|$ such that the following conditions are satisfied:

- (i) Each T_i is a perpetual directed trace.
- (ii) For each $i, j \in \{1, \dots, m\}$, if $i \neq j$ then $E_i \cap E_j = \emptyset$ and $\text{alph}(T_i) \times \text{alph}(T_j) \subseteq I$.
- (iii) $E = \cup_{i=1}^m E_i$, $\leq = \cup_{i=1}^m \leq_i$ and $\lambda = \cup_{i=1}^m \lambda_i$.

Proof

Let $T = (E, \leq, \lambda)$ be a perpetual trace and let $D_T = (\text{alph}(T) \times \text{alph}(T)) \cap D$, i.e., D_T is the dependency relation restricted to the letters from T . Define a binary relation $\uparrow \subseteq E \times E$ via:

$$e \uparrow e' \quad \text{iff} \quad \exists e''. \quad e \leq e'' \text{ and } e' \leq e''. \quad (2)$$

We wish to show that \uparrow is an equivalence relation. For this we will need three observations.

Observation 11.1 Suppose $(a, b) \in D_T$ and $e \in E$ with $\lambda(e) = a$. Then there exists $e' \geq e$ with $\lambda(e') = b$.

To see this, note that as T is perpetual, there must exist infinitely many events labelled by b . For each such event e_b we have $e_b \leq e$ or $e \leq e_b$ by condition T3 in the definition of a trace. It cannot be the case that all these events are \leq -smaller than e ; this would contradict the condition (T1) of the definition of a trace. Hence there is an event e' labelled by b that is not \leq -smaller than e . By the condition (T3) we have: $e \leq e'$.

Observation 11.2 Let $e, e' \in E$ with $e < e'$. Then $(\lambda(e), \lambda(e')) \in D_T^*$; where D_T^* is the reflexive and transitive closure of the relation D_T

Call a *path* from e to e' in T a sequence $e = e_0 < e_1 < \dots < e_n = e'$. Clearly such a path must exist because $e < e'$. This follows from the condition (T1) in the definition of a trace. Again, by condition (T2) in the definition of a trace, we have $(\lambda(e_i), \lambda(e_{i+1})) \in D_T$ for $0 \leq i < n$.

Observation 11.3 For every $e, e' \in E$ we have: $e \uparrow e'$ iff $(\lambda(e), \lambda(e')) \in D_T^*$

If this holds then \uparrow is an equivalence relation because D_T^* is an equivalence relation. To establish the observation first assume that $e'' \in E$ with $e \leq e''$ and $e' \leq e''$ so that $e \uparrow e'$. From Observation 11.2 and the fact that D_T^* is an equivalence relation, we at once have $(\lambda(e), \lambda(e')) \in D_T^*$. Conversely assume that $(\lambda(e), \lambda(e')) \in D_T^*$ with $\lambda(e) = a$ and $\lambda(e') = b$. If $a = b$ then $e \uparrow e'$ follows at once from condition (T3) in the definition of a trace. So assume $a \neq b$. Let a_0, a_1, \dots, a_n be a sequence such that $a = a_0$, $a_n = b$ and $(a_i, a_{i+1}) \in D_T$ for $0 \leq i < n$. By repeated applications of Observation 11.1 we can find a sequence of events e_0, e_1, \dots, e_n in E such that $e = e_0$, $\lambda(e_i) = a_i$ and $e_i \leq e_{i+1}$ for $0 \leq i < n$. Since $\lambda(e_n) = b = \lambda(e')$ we must have $e' \leq e_n$ or $e_n \leq e'$. In either case, $e \uparrow e'$ as required.

To finish the proof of the proposition, let $\{eq_1, eq_2, \dots, eq_m\}$ be the set of D_T^* -equivalence classes of $\text{alph}(T)$. Define $T_i = (T|eq_i, \leq|eq_i, \lambda|eq_i)$ where $|eq_i$ denotes the restriction to the events labelled with the letters in eq_i . Conditions (i) and (ii) follow from Observation 11.3. Condition (iii) follows directly from the definition of the traces T_i . \square

We conclude by defining the crucial notion of shapes.

Definition 12 (Shape) (i) The *shape* of a perpetual trace T is the family $\{\text{alph}(T_i)\}_{i=1}^m$ where $\{T_i\}_{i=1}^m$ is the decomposition described above. (In other words the shape of T is the set of alphabets of D_T^* -equivalence classes of T)

(ii) A family $\{\Sigma_i\}_{i=1}^m$ is a *shape in the trace alphabet* (Σ, I) if it is the shape of some perpetual trace over this alphabet.

7 Normal linearizations of traces

Our goal here is to define a *partial* linearization map whose domain will be the set of finite and perpetual directed traces. This partial map will have the property that it can be captured (in the sense of Definition 7) by a fixed formula in $LTrL$.

Through the rest of the section we fix a strict linear order $\prec \subseteq \Sigma \times \Sigma$. For $\emptyset \neq \Sigma' \subseteq \Sigma$, $\min(\Sigma')$ will denote the least element of Σ' under \prec .

Let $T = (E, \leq, \lambda)$ be a trace. Then the relation $co \subseteq E \times E$ is defined as: $e \text{ co } e'$ iff $e \not\leq e'$ and $e' \not\leq e$. Further, for $e, e' \in E$ we set $\Sigma_{ee'} = \lambda(\uparrow e - \uparrow e')$.

(For $X \subseteq E$, $\lambda(X) = \{\lambda(x) \mid x \in X\}$.)

Definition 13 Let $T = (E, \leq, \lambda)$ be a trace. Then $lex_T \subseteq E \times E$ is defined as: $e lex_T e'$ iff one of the two following conditions is satisfied.

- (i) $e < e'$
- (ii) $e co e'$ and $\min(\Sigma_{ee'}) \prec \min(\Sigma_{e'e})$.

Suppose $T = (E, \leq, \lambda)$ is a trace and $e, e' \in E$ with $e co e'$. Then it is easy to show that $\Sigma_{ee'} \cap \Sigma_{e'e} = \emptyset$ and that both $\Sigma_{ee'}$ and $\Sigma_{e'e}$ are nonempty. Hence lex_T is well-defined.

Lemma 14 Let $T = (E, \leq, \lambda)$ be a trace. Then (E, lex_T) is a strict linear order.

Proof

Let $e, e' \in E$ with $e \neq e'$. It is straightforward to verify that $e lex_T e'$ or $e' lex_T e$ but not both. So we need to show that lex_T is transitive.

Let $e_1, e_2, e_3 \in E$ with $e_1 lex_T e_2$ and $e_2 lex_T e_3$. To show $e_1 lex_T e_3$, first note that e_1, e_2 and e_3 must be pairwise distinct. For distinct $i, j \in \{1, 2, 3\}$ we fix (if it exists) an event $e_{ij} \in \uparrow e_i - \uparrow e_j$ labelled with $\min(\Sigma_{e_i e_j})$. We need to examine several, quite easy, cases.

Suppose $e_1 < e_2$. Then $\uparrow e_2 - \uparrow e_3 \subseteq \uparrow e_1 - \uparrow e_3$ and $\uparrow e_3 - \uparrow e_1 \subseteq \uparrow e_3 - \uparrow e_2$. As $lex_T(e_2, e_3)$ we get $lex_T(e_1, e_3)$.

The case when $e_2 \leq e_3$ is done similarly. If $e_1 \leq e_3$ then $lex_T(e_1, e_3)$ and we are done.

Suppose $e_1 co e_2$ and $e_2 co e_3$ and $e_1 \not\leq e_3$. We claim that $e_1 co e_3$. If it were $e_3 \leq e_1$ then $\uparrow e_1 - \uparrow e_2 \subseteq \uparrow e_3 - \uparrow e_2$ and $\uparrow e_2 - \uparrow e_3 \subseteq \uparrow e_2 - \uparrow e_1$. Hence $\lambda(e_{32}) \preceq \lambda(e_{12})$ and $\lambda(e_{21}) \preceq \lambda(e_{23})$. We also know that $\lambda(e_{12}) \prec \lambda(e_{21})$. This gives us $\lambda(e_{32}) \prec \lambda(e_{23})$, a contradiction.

Hence we are left with the case when e_1, e_2, e_3 are pairwise in co relation. From $lex_T(e_1, e_2)$ and $lex_T(e_2, e_3)$ we get $\lambda(e_{12}) \prec \lambda(e_{21})$ and $\lambda(e_{23}) \prec \lambda(e_{32})$.

First we claim that:

$$\lambda(e_{13}) \preceq \lambda(e_{12}). \tag{3}$$

Suppose $e_{12} \notin \uparrow e_3$. Then $e_{12} \in \uparrow e_1 - \uparrow e_3$ and (3) follows. So assume that $e_{12} \in \uparrow e_3$. Then $e_{12} \in \uparrow e_3 - \uparrow e_2$. Since $e_2 \text{ lex}_T e_3$ we have:

$$\lambda(e_{23}) \prec \lambda(e_{32}) \preceq \lambda(e_{12}). \quad (4)$$

Now we must consider two cases. Suppose $e_{23} \in \uparrow e_1$. Then $e_{23} \in \uparrow e_1 - \uparrow e_3$ and hence $\lambda(e_{13}) \preceq \lambda(e_{23})$ which then leads to (3). Suppose on the other hand $e_{23} \notin \uparrow e_1$. Then $e_{23} \in \uparrow e_2 - \uparrow e_1$ which leads to $\lambda(e_{12}) \prec \lambda(e_{21}) \preceq \lambda(e_{23})$. But from (4) above we now have the contradiction: $\lambda(e_{12}) \prec \lambda(e_{12})$. Hence (3) must hold.

To finish the proof there are two cases to consider. Suppose $e_{31} \in \uparrow e_2$. Then $e_{31} \in \uparrow e_2 - \uparrow e_1$ and from $\lambda(e_{12}) \prec \lambda(e_{21}) \preceq \lambda(e_{31})$ and (3) we can deduce $\lambda(e_{13}) \prec \lambda(e_{31})$. So suppose that $e_{31} \notin \uparrow e_2$. Then $e_{31} \in \uparrow e_3 - \uparrow e_2$ and consequently $\lambda(e_{23}) \prec \lambda(e_{32}) \preceq \lambda(e_{31})$. If $e_{23} \in \uparrow e_1$ then $e_{23} \in \uparrow e_1 - \uparrow e_3$ and hence $\lambda(e_{13}) \preceq \lambda(e_{23}) \prec \lambda(e_{31})$ as desired. If on the other hand, $e_{23} \notin \uparrow e_1$ then $e_{23} \in \uparrow e_2 - \uparrow e_1$ and hence $\lambda(e_{12}) \prec \lambda(e_{21}) \preceq \lambda(e_{23})$. This in turn leads to $\lambda(e_{12}) \prec \lambda(e_{31})$. From (3) we can again conclude that $\lambda(e_{13}) \prec \lambda(e_{31})$. \square

We shall introduce the notion of normal configurations that in turn will enable us to define normal linearisations of traces.

Definition 15 (Normal linearization) (i) Let $T = (E, \leq, \lambda)$ be a trace and $c \in C_T$. Then c is a *normal configuration* iff c is lex_T closed, i.e., for every $e \in c$ and every $e' \in E$, if $e' \text{ lex}_T e$ then $e' \in c$.

(ii) Let σ be a linearisation of T with ρ as the run map of σ (as defined in Section 3). Then σ is a *normal linearisation* of T iff $\rho(\tau)$ is a normal configuration for every $\tau \in \text{prf}(\sigma)$.

It is easy to see that there can be at most one normal linearisation of a trace. Some traces do not have normal linearisations. One of the reasons why we focus on directed perpetual traces is:

Lemma 16 If T is finite or a directed perpetual trace then there exists a unique normal linearisation of T .

Proof

Let c be a configuration of $T = (E, \leq, \lambda)$. We say that the event $e \in E$ is

enabled at c iff $e \notin c$ and $c \cup \{e\}$ is a configuration. It is easy to see that e is enabled at c iff e is a minimal element of $E - c$ under \leq . Next we note that if c is a normal configuration of the trace T and e is the least enabled event at c under lex_T (among all the enabled events at c), then $c \cup \{e\}$ is also a normal configuration. From the fact that the empty configuration is always normal, it now follows that if T is a finite trace then it admits a unique normal linearisation.

One can apply the same reasoning in the case of directed perpetual traces but it may be not clear that the obtained sequence contains all the events of the trace. To show that it is indeed the case it is enough to show that for every event e the set $\{e' \mid e' lex_T e\}$ is finite.

Let $\{a_1, \dots, a_k\} = alph(T)$. Take an event $e_1 \geq e$ labelled with a_1 . Such an event exists because T is directed and perpetual. Then inductively for every $i = 2, \dots, k$ take $e_i \geq e_{i-1}$ labelled by a_i . We claim that if $e' lex_T e$ then $e' \leq e_k$. If $e' \leq e$ then it is obvious. If $e' co e$ then let a_j be the label of e' . Clearly $e' \leq e_j$. Hence $e' \leq e_k$. We are done, as the set $\downarrow e_k$ is finite. \square

Thus the partial linearization map nrl we have in mind is defined only for finite and perpetual directed traces. For each such trace T , we define $nrl(T)$ to be the normal linearization of T .

8 Defining normal linearizations in $LTrL(\Sigma, I)$

Here we construct a formula NRC which will capture the (partial) linearization map nrl defined in the previous section. This is done in lemma 17. Although the lemma is stated for all traces, it does not mean that the formula NRC will work for all traces. The catch is that the notion of a normal configuration makes sense for any trace T since it is defined in terms of lex_T . However, for an infinite trace T , the element of Σ^ω induced by lex_T may not be a linearisation of T ! Fortunately we know that by virtue of Lemma 16, NRC will have the desired effect when applied to finite and perpetual directed traces.

Lemma 17 There exists an $LTrL(\Sigma, I)$ formula NRC such that for every $T \in TR$ and every $c \in C_T$: $T, c \models NRC$ iff c is a normal configuration.

Proof

We will say that the event e is at the *top* of the configuration c iff $c \cap \uparrow e = \{e\}$. In other words, e is a maximal element of c under \leq . We let $top(c)$ be the set of elements that are on the top of c . Recall that we say that an event e is enabled in a configuration c if $e \notin c$ and $c \cup \{e\}$ is a configuration.

Observation 17.1 Let $T = (E, \leq, \lambda)$ be a trace and $c \in C_T$. Then c is *not* a normal configuration iff there exist events e, e' and e_1 satisfying the following conditions:

- (i) $e \in top(c)$, e' is enabled at c and $e_1 \in \uparrow e' - \uparrow e$,
- (ii) $\forall e_2 \in E$, if $e_2 \in \uparrow e - \uparrow e'$ then $\lambda(e_1) \prec \lambda(e_2)$.

To see that this holds assume first that c is not a normal configuration. Then there exists $e_3 \in c$ and $e'_3 \notin c$ such that $e'_3 \text{ lex}_T e_3$. Let $e \in top(c)$ such that $e_3 \leq e$ and let e' be enabled at c such that $e' \leq e'_3$. By the transitivity of lex_T we now have $e' \text{ lex}_T e$. Let $x = \min(\Sigma_{e'e})$ and $e_1 \in \uparrow e' - \uparrow e$ such that $\lambda(e_1) = x$. Now suppose $e_2 \in \uparrow e - \uparrow e'$. By the definition of lex_T , we have $x \prec \lambda(e_2)$.

Next suppose there exist e, e' and e_1 fulfilling the conditions specified by Observation 17.1. Let $e_2 \in \uparrow e - \uparrow e'$. Then $\lambda(e_1) \prec \lambda(e_2)$. Hence $\min(\Sigma_{e'e}) \preceq \lambda(e_1) \prec \min(\Sigma_{e'e})$. Thus $e' \text{ lex}_T e$ and c is not normal.

We need to define an intermediate formula before getting to *NRC*. In what follows, for $a, b, d \in \Sigma$ let $\Gamma_{ab}^d = \{S \subseteq \Sigma : a \in S \text{ and } b, d \notin S\}$. We will use this notion only in contexts where $a \neq b$ and $a \neq d$.

For $a, b, d \in \Sigma$, define the formula μ_{ab}^d to be $\sim \underline{tt}$ in case $a = b$ or $a = d$. Otherwise,

$$\mu_{ab}^d = \langle b^{-1} \rangle \underline{tt} \wedge \bigvee_{S \in \Gamma_{ab}^d} \alpha_S$$

where

$$\alpha_S = \left(\bigwedge_{x \in S} \langle x^{-1} \rangle \underline{tt} \right) U \left(\langle d^{-1} \rangle \underline{tt} \wedge \bigwedge_{x \in S} \langle x^{-1} \rangle \underline{tt} \wedge \bigwedge_{y \in (\Sigma - S) - \{d\}} \sim \langle y^{-1} \rangle \underline{tt} \right).$$

The meaning of the formula μ_{ab}^d is the best described by the observation below. We have also tried to present it graphically (Figure 1). Horizontal line represents the top of the configuration. The thick part represents those

events from the top which labels go to the set S in α_S . Dashed cone from e' represents all the events in the future of e' . A downwards cone from e_1 represents all the events from the past of e_1 .

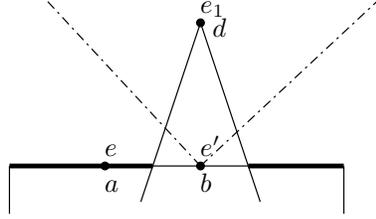


Figure 1: The meaning of the formula μ_{ab}^d

Observation 17.2 Let $T = (E, \leq, \lambda)$ be a trace and $c \in C_T$. Then $T, c \models \mu_{ab}^d$ iff there exist events e, e', e_1 such that the following conditions are satisfied:

- (i) $\lambda(e) = a, \lambda(e') = b$ and $\lambda(e_1) = d$.
- (ii) $e, e' \in \text{top}(c)$ with $e \neq e'$ and $e_1 \in \uparrow e' - \uparrow e$.

To see that this must hold first suppose that $T, c \models \mu_{ab}^d$. Then $a \neq b$ and $a \neq d$. Let $S \in \Gamma_{ab}^d$ such that $T, c \models \langle b^{-1} \rangle \underline{tt} \wedge \alpha_S$. Then there exists c' such that $c \subseteq c'$ and

$$T, c' \models \langle d^{-1} \rangle \underline{tt} \wedge \bigwedge_{x \in S} \langle x^{-1} \rangle \underline{tt} \wedge \bigwedge_{y \in (\Sigma - S) - \{d\}} \sim \langle y^{-1} \rangle \underline{tt}. \quad (1)$$

Furthermore, for every configuration c'' such that $c \subseteq c'' \subset c'$, we have

$$T, c'' \models \bigwedge_{x \in S} \langle x^{-1} \rangle \underline{tt}. \quad (2)$$

Let $e, e' \in \text{top}(c)$ such that $\lambda(e) = a$ and $\lambda(e') = b$. Clearly, $a \neq b$ implies $e \neq e'$.

Now suppose $b = d$. Then by setting $e_1 = e'$, we at once get the desired conclusion. This follows from the fact that $a \neq b$ because $b \notin S$ and hence $\lambda(e) \neq \lambda(e')$. But then $e, e' \in \text{top}(c)$ and thus $e' \in \uparrow e' - \uparrow e$.

So assume that $b \neq d$. Then $T, c' \models \sim \langle b^{-1} \rangle \underline{tt}$ because $b \in (\Sigma - S) - \{d\}$.

Let $e_1 \in \text{top}(c')$ such that $\lambda(e_1) = d$. Such an e_1 must exist because $T, c' \models \langle d^{-1} \rangle \underline{tt}$. We now wish to argue that e, e' and e_1 have the desired properties.

Let $S = \{a^1, a^2, \dots, a^k\}$. Note that $a \in S$. Since $T, c \models \bigwedge_{x \in S} \langle x^{-1} \rangle \underline{tt}$, we can fix $e^1, e^2, \dots, e^k \in \text{top}(c)$ such that $\lambda(e^j) = a^j$ for each $j \in \{1, 2, \dots, k\}$. Clearly $e \in \{e^1, \dots, e^j\}$. We will first argue that $e^j \text{ co } e_1$ for every j which will lead to $e \text{ co } e_1$. So fix $j \in \{1, 2, \dots, k\}$ and suppose that $e^j \text{ co } e_1$ does not hold. Since $e_1 \in \text{top}(c')$ and $c \subseteq c'$ and $e^j \in \text{top}(c)$ we can rule out $e_1 < e^j$. So it must be the case that $e^j < e_1$. But this implies that there exists a finite chain $e^j = z_0 < z_1 < \dots < z_n = e_1$. Since $\lambda(e_1) = d \notin S$ we have $\lambda(e^j) \neq d$ and $n \geq 1$. Let i be the least integer in $\{0, 1, \dots, n-1\}$ such that $\lambda(z_i) = a^j$ and $\lambda(z_{i+1}) \neq a^j$. Let $\lambda(z_{i+1}) = \hat{a}$. Clearly $a^j D \hat{a}$. Now consider the configuration $\hat{c} = c \cup \downarrow z_{i+1}$. It is easy to check that $c \subseteq \hat{c} \subseteq c'$. Hence $T, \hat{c} \models \bigwedge_{x \in S} \langle x^{-1} \rangle \underline{tt}$ which then implies $T, \hat{c} \models \langle (a^j)^{-1} \rangle \underline{tt}$. But $z_{i+1} \in \text{top}(\hat{c})$ and hence $T, \hat{c} \models \langle (\hat{a})^{-1} \rangle \underline{tt}$. This is a contradiction because two distinct labels at the top of a configuration can not be in the dependence relation. Thus $e^j \text{ co } e_1$ and consequently $e \text{ co } e_1$.

Next we must show that $e' \leq e_1$. Since $T, c \models \langle b^{-1} \rangle \underline{tt}$ and $T, c' \models \sim \langle b^{-1} \rangle \underline{tt}$ (recall that we are considering the case $b \neq d$) we know that $e' \notin \text{top}(c')$. Hence there exists $e'' \in \text{top}(c')$ such that $e' < e''$. If $e'' = e_1$, we are done. Otherwise $e' < e''$ for some $e'' \in \text{top}(c')$ with $\lambda(e'') = a^j$ for some $a^j \in S$. We will now argue that this is impossible.

Suppose $e' < e''$ and $\lambda(e'') = a^j \in S$ with $e'' \in \text{top}(c')$. Then from $e^j \in \text{top}(c)$ and $b \notin S$, we get $a^j I b$. Consequently $a^j \neq b$. Clearly there exists a non-null path $e' = z_0 < z_1 < \dots < z_n = e''$. Let i be the largest integer in $\{1, 2, \dots, n\}$ such that $z_i = a^j$ and $z_{i-1} \neq a^j$. Let $\lambda(z_{i-1}) = \hat{a}$ and $\hat{c} = c \cup \downarrow z_{i-1}$. It is easy to check that $c \subseteq \hat{c} \subseteq c'$ and hence $T, \hat{c} \models \langle (a^j)^{-1} \rangle \underline{tt}$. But $z_{i-1} \in \text{top}(\hat{c})$ and hence $T, \hat{c} \models \langle (\hat{a})^{-1} \rangle \underline{tt}$. We now have a contradiction because $\hat{a} \neq a^j$ and $\hat{a} D a^j$.

To prove the right to left implication of Observation 17.2 assume that the event e, e' and e_1 exist which fulfill the properties specified in the observation. Let $c' = c \cup \downarrow e_1$. Then $e_1 \in \text{top}(c')$ and hence $T, c' \models \langle d^{-1} \rangle \underline{tt}$. Let $S = \{\lambda(e'') \mid e'' \in \text{top}(c') \text{ and } e'' \neq e_1\}$. First we assert $a \in S$. This is because $e \in \text{top}(c)$ and $e \text{ co } e_1$. Hence $e \in \text{top}(c')$ as well because $c' = c \cup \downarrow e_1$. By the definition of S we are assured that $d \notin S$. Hence $a \neq d$. Next suppose

$b \in S$. Then there exists $e'' \in \text{top}(c')$ such that $e'' \neq e_1$ and $\lambda(e'') = b$. But then $e'' \in c \cup \downarrow e_1$ and since $e'' \neq e_1$ implies $e'' \text{ co } e_1$, we must have $e'' \in c$. In fact $e'' \in \text{top}(c)$ because $e'' \in \text{top}(c')$ and $c \subseteq c'$. But this implies $e'' = e'$ which contradicts $e' \leq e_1$. Thus $b \notin S$ and consequently $b \neq a$.

Clearly, by the choice of S , we have

$$T, c' \models \langle d^{-1} \rangle \underline{tt} \wedge \bigwedge_{x \in S} \langle x^{-1} \rangle \underline{tt} \wedge \bigwedge_{y \in (\Sigma - S) - \{d\}} \sim \langle y^{-1} \rangle \underline{tt}.$$

It is also clear that $T, c \models \langle b^{-1} \rangle \underline{tt}$. So suppose $c \subseteq c'' \subset c'$ and $\hat{a} \in S$. Then there exists $e'' \in \text{top}(c')$ such that $\lambda(e'') = \hat{a}$. But then $c' = c \cup \downarrow e_1$ and $d \notin S$ at once leads to $\hat{a} \in \text{top}(c'')$ as well. Hence $T, c'' \models \langle (\hat{a}^{-1}) \rangle \underline{tt}$. We now have $T, c \models \mu_{ab}^d$.

Now we define the desired formula NRC as:

$$NRC \approx \sim \bigvee_{(a,b) \in I} \langle b \rangle \left(\bigvee_{d \in \Sigma} \left(\mu_{ab}^d \wedge \bigwedge_{d' \prec d} \sim \mu_{ba}^{d'} \right) \right).$$

To see that NRC has the required property assume first that $T = (E, \leq, \lambda)$ is a trace and $\hat{c} \in C_T$ is a configuration that is not normal. Then by Observation 17.1, there exist event e, e' and e_1 such that $e \in \text{top}(\hat{c})$, e' is enabled at \hat{c} and $e_1 \in \uparrow e' - \uparrow e$. Further, if $e_2 \in \uparrow e - \uparrow e'$ then $\lambda(e_1) \prec \lambda(e_2)$. Let $\lambda(e) = a$, $\lambda(e') = b$ and $\lambda(e_1) = d$. If $e \leq e'$ then this would lead to $e \leq e_1$ contradicting $e \text{ co } e_1$. Hence $e \text{ co } e'$ as well. Consequently $a \neq b$ and $a \neq d$. Now consider the configuration $c = \hat{c} \cup \{e'\}$. Clearly c fulfills the requirements of Observation 17.2 and hence $T, c \models \mu_{ab}^d$. Now suppose $T, c \models \mu_{ba}^{d'}$ for some $d' \prec d$. Then by the definition of the formula $\mu_{ba}^{d'}$ we are assured that $b \neq d'$. Further, we already have $b \neq a$. Now again by Observation 17.2, there exists e_2 such that $e_2 \in \uparrow e - \uparrow e'$ with $\lambda(e_2) \prec \lambda(e_1)$. But this contradicts the criteria justifying the choice of e, e' and e_1 . Hence $T, \hat{c} \models \sim NRC$.

Next suppose $T, \hat{c} \models \sim NRC$. Then there exists $(a, b) \in I$ and $d \in \Sigma$ such that $T, \hat{c} \models \langle b \rangle \left(\mu_{ab}^d \wedge \bigwedge_{d' \prec d} \sim \mu_{ba}^{d'} \right)$. Clearly $a \neq b$ and $a \neq d$. Hence there exists an event $e \in \text{top}(\hat{c})$ and an event e' which is enabled at \hat{c} such that $\lambda(e) = a$ and $\lambda(e') = b$. Moreover with $c = \hat{c} \cup \{e'\}$, we have $T, c \models \mu_{ab}^d \wedge \bigwedge_{d' \prec d} \sim \mu_{ba}^{d'}$. Because $T, c \models \mu_{ab}^d$ there exists an event e_1 such that $\lambda(e_1) = d$ and $e_1 \in \uparrow e' - \uparrow e$. This follows from Observation 17.2. Now suppose there exists $e_2 \in \uparrow e - \uparrow e'$ such that $\lambda(e_2) = d' \prec d$. If $d' \neq b$ then,

by Observation 17.2, we have $T, c' \models \mu_{ba}^{d'}$; a contradiction. Hence it must be the case that $d' = b$ so that $\mu_{ba}^{d'} = \sim \underline{tt}$. But this is again a contradiction, because $\lambda(e_2) = d' = b$ implies that $e' \leq e_2$ or $e_2 \leq e'$ whereas we are supposed to have e_2 *co* e' . Thus $\min(\Sigma_{e'e}) \preceq d \prec \min(\Sigma_{ee'})$. This leads to $e' \text{ lex}_T e$, which then guarantees that \hat{c} is not a normal configuration. \square

9 Decomposing formulas in $FO(\Sigma, I)$

In this section we show how to decompose a formula in $FO(\Sigma, I)$ into conjunctions of formulas that talk separately about the finite and perpetual directed components of a trace. From Proposition 10 and Proposition 11 we know that every trace can be canonically decomposed into a finite part and a bounded number perpetual directed parts. Further, using the formula *NRC* constructed in the previous section, we can apply Lemmas 6 and 8 to translate an $FO(\Sigma, I)$ sentence ψ into an *LTrL* formula α such that ψ and α define the same language of finite and perpetual directed traces. Thus at the end of this section, we will be able to translate any $FO(\Sigma, I)$ sentence into conjunctions of *LTrL* assertions such that any trace T is a model of ψ iff the finite and perpetual directed components of T satisfy the *LTrL* assertions obtained from ψ . In the next section we will show how these *LTrL* formulas can be put together to form a single *LTrL* formula.

The decomposition result we are after is an easy consequence of the decomposition theorem of Feferman and Vaught [7]. For the sake of completeness we recall this theorem here. The reader familiar with this topic can proceed directly to Lemma 19.

Let us fix some finite relational signature $\text{Sig} = \{R_1, \dots, R_l\}$. Given two structures $\mathcal{A} = \langle A, R_1^A, \dots, R_l^A \rangle$ and $\mathcal{B} = \langle B, R_1^B, \dots, R_l^B \rangle$ of this signature we define their *disjoint union* as the structure $\mathcal{A} \oplus \mathcal{B}$ of the signature $\text{Sig} \cup \{in_1, in_2\}$:

$$\mathcal{A} \oplus \mathcal{B} = \langle A \oplus B, R_1^A \oplus R_1^B, \dots, R_l^A \oplus R_l^B, in_1^{A \oplus B}, in_2^{A \oplus B} \rangle$$

here $A \oplus B$ and $R_i^A \oplus R_i^B$ stand for disjoint sums of the appropriate sets and $in_1^{A \oplus B}(a)$ holds if $a \in A$. Similarly $in_2^{A \oplus B}(b)$ holds if $b \in B$.

Theorem 18 (Composition thm. for disjoint sum)

Let Sig be a finite relational signature. Let φ be a sentence of $FO(\text{Sig} \cup \{in_1, in_2\})$.

There exists a finite collection of pairs $(\psi_1, \psi'_1), \dots, (\psi_k, \psi'_k)$ of $FO(\text{Sig})$ sentences, such that, for every two structures \mathcal{A}, \mathcal{B} of the signature Sig we have:

$$\mathcal{A} \oplus \mathcal{B} \models \varphi \text{ iff there exists } i \in \{1, 2, \dots, k\} \text{ with } \mathcal{A} \models \psi_i \text{ and } \mathcal{B} \models \psi'_i.$$

Proof

The proof is a standard application of Ehrenfeucht-Fraïssé games. For a description of the games see for example [4]. We denote the n -move game on structures \mathcal{A} and \mathcal{B} by $G_n(\mathcal{A}, \mathcal{B})$. Let us denote by $\text{qd}(\theta)$ the *quantifier depth* of the sentence θ . We define an n -theory of a structure \mathcal{C} as the set of sentences $\text{Th}_n(\mathcal{C}) = \{\theta : \text{qd}(\theta) \leq n \text{ and } \mathcal{C} \models \theta\}$. We have the following characterisation of n -theories in terms of Ehrenfeucht-Fraïssé games.

Observation 18.1 Two structures \mathcal{A}, \mathcal{B} have the same n -theories iff Duplicator has a winning strategy in the n -move Ehrenfeucht-Fraïssé game. Every n -theory is equivalent to a single sentence, i.e., for every n -theory Γ there exist a sentence θ_Γ such that for every structure \mathcal{A} : $\text{Th}_n(\mathcal{A}) = \Gamma$ iff $\mathcal{A} \models \theta_\Gamma$.

The proof of this observation relies on the fact that the signatures are finite and relational.

The next observation is that the n -theory of $\mathcal{A} \oplus \mathcal{B}$ is determined by the n -theories of \mathcal{A} and \mathcal{B} . Indeed suppose that $\text{Th}_n(\mathcal{A}) = \text{Th}_n(\mathcal{A}')$ and $\text{Th}_n(\mathcal{B}) = \text{Th}_n(\mathcal{B}')$. By Observation 18.1 it is enough to show that Duplicator has a winning strategy in the n -move game $G_n(\mathcal{A} \oplus \mathcal{B}, \mathcal{A}' \oplus \mathcal{B}')$. By assumption Duplicator has winning strategies in the games $G_n(\mathcal{A}, \mathcal{A}')$ and $G_n(\mathcal{B}, \mathcal{B}')$. The strategy in $G_n(\mathcal{A} \oplus \mathcal{B}, \mathcal{A}' \oplus \mathcal{B}')$ is to copy moves of Spoiler in this game to $G_n(\mathcal{A}, \mathcal{A}')$ or $G_n(\mathcal{B}, \mathcal{B}')$ and consult the strategies there. For example if Spoiler puts a pebble on some element of the \mathcal{A} component of $\mathcal{A} \oplus \mathcal{B}$ then we put Spoilers pebble on the same element in the game $G_n(\mathcal{A}, \mathcal{A}')$. The winning strategy of Duplicator in $G_n(\mathcal{A}, \mathcal{A}')$ puts a pebble on some element of \mathcal{A}' . We copy this move by putting a pebble on the same element of the \mathcal{A}' component of $\mathcal{A}' \oplus \mathcal{B}'$. It should be clear that such a strategy is winning for Duplicator.

After these preliminary remarks we are ready to prove the theorem. Let φ be a $FO(\text{Sig} \cup \{in_1, in_2\})$ sentence. Let n be the quantifier depth of φ .

Let $(\Gamma_1, \Gamma'_1), \dots, (\Gamma_k, \Gamma'_k)$ be all the pairs of n -theories such that:

$$\text{if } \text{Th}_n(\mathcal{A}) = \Gamma_i \text{ and } \text{Th}_n(\mathcal{B}) = \Gamma'_i \text{ then } \varphi \in \text{Th}_n(\mathcal{A} \oplus \mathcal{B}) \quad (5)$$

By a simple induction on n one can show that there are finitely many n -theories. Hence, the number of pairs as in (5) is finite. From Observation 18.1 we know that for every Γ_i there exists a formula ψ_i , such that, for every structure \mathcal{A} : $\text{Th}_n(\mathcal{A}) = \Gamma_i$ iff $\mathcal{A} \models \psi_i$. Similarly for every Γ'_i we can find ψ'_i . We claim that $(\psi_1, \psi'_1), \dots, (\psi_k, \psi'_k)$ satisfies the statement of the theorem.

For the right to left implication suppose $\mathcal{A} \models \psi_i$ and $\mathcal{B} \models \psi'_i$ then by (5) we have $\varphi \in \text{Th}_n(\mathcal{A} \oplus \mathcal{B})$. For the left to right implication suppose that $\mathcal{A} \oplus \mathcal{B} \models \varphi$. Then $\varphi \in \text{Th}_n(\mathcal{A} \oplus \mathcal{B})$. The n -theory of $\mathcal{A} \oplus \mathcal{B}$ is determined by the n -theories of \mathcal{A} and \mathcal{B} . Hence there exists i , s.t. $\text{Th}_n(\mathcal{A}) = \Gamma_i$ and $\text{Th}_n(\mathcal{B}) = \Gamma'_i$. So $\mathcal{A} \models \psi_i$ and $\mathcal{B} \models \psi'_i$. \square

Let us now come back to decomposing traces. First we show that we can separate finite and infinite part.

Lemma 19 Let $\varphi \in FO(\Sigma, I)$. There exists a finite collection of pairs of $FO(\Sigma, I)$ formulas, $(\psi_1, \psi'_1), (\psi_2, \psi'_2), \dots, (\psi_k, \psi'_k)$, such that, for every $T \in TR^{inf}$: $T \models^{FO} \varphi$ iff there is $i \in \{1, 2, \dots, k\}$ with $\text{fin}(T) \models^{FO} \psi_i$ and $\text{inf}(T) \models^{FO} \psi'_i$.

Proof

Let $\varphi \in FO(\Sigma, I)$ be given. We claim that there exists a formula φ' , such that, for every infinite trace T :

$$T \models^{FO} \varphi \quad \text{iff} \quad \text{fin}(T) \oplus \text{inf}(T) \models \varphi' \quad (6)$$

For this we show that in $\text{fin}(T) \oplus \text{inf}(T)$ we can recover the ordering of T by means of a first order formula. Recall that $\text{fin}(T) \oplus \text{inf}(T)$ is a structure of a signature $\{R_a\}_{a \in \Sigma} \cup \{\leq, in_1, in_2\}$. The carriers of T and $\text{fin}(T) \oplus \text{inf}(T)$ are the same. Also the interpretations of the relations $\{R_a\}_{a \in \Sigma}$ are the same. The interpretation of \leq relation in $\text{fin}(T) \oplus \text{inf}(T)$ is the (disjoint) union of \leq_{fin} and \leq_{inf} where $\text{fin}(T) = (E_{\text{fin}}, \leq_{\text{fin}}, \lambda_{\text{fin}})$ and $\text{inf}(T) = (E_{\text{inf}}, \leq_{\text{inf}}, \lambda_{\text{inf}})$.

Consider the formula:

$$\begin{aligned} \theta(x, y) = & \left(in_1(x) \wedge in_1(y) \wedge x \leq y \right) \vee \left(in_2(x) \wedge in_2(y) \wedge x \leq y \right) \\ & \vee \left(in_1(x) \wedge in_2(y) \wedge \exists z_1 \exists z_2. in_1(z_1) \right. \\ & \left. \wedge in_2(z_2) \wedge D(z_1, z_2) \wedge x \leq z_1 \wedge z_2 \leq y \right) \end{aligned}$$

where $D(z_1, z_2)$ is a formula stating that the labels of z_1 and z_2 are dependent. It is not difficult to check that for all nodes x, y of T we have: $T \models x \leq y$ iff $\text{fin}(T) \oplus \text{inf}(T) \models \theta(x, y)$. Hence taking φ and replacing all subformulas of the form $x \leq y$ by $\theta(x, y)$ we obtain a formula φ' satisfying the condition (6). The lemma now follows directly from Theorem 18. \square

Next we further break up the assertions concerning $\text{inf}(T)$ to mimic the decomposition described in Proposition 11.

Lemma 20 Let $\varphi \in FO(\Sigma, I)$ and $sh = \{\Sigma_i\}_{i=1}^m$ be a shape of (Σ, I) (cf. Definition 12). Then there exists a finite array of formulas

$$(\theta_1^1, \dots, \theta_m^1), (\theta_1^2, \dots, \theta_m^2), \dots, (\theta_1^n, \dots, \theta_m^n)$$

such that the following conditions are satisfied:

- (i) $\theta_i^j \in FO(\Sigma_i, I_i)$ for every $i \in \{1, 2, \dots, m\}$ and every $j \in \{1, 2, \dots, n\}$. (Observe that the formulas with different subscripts have disjoint alphabets. As might be expected, $I_i = (\Sigma_i \times \Sigma_i) \cap I$)
- (ii) Suppose $T \in TR^{mf}$, and $\text{inf}(T)$ is of shape sh . Let $\{T_i\}_{i=1}^m$ be a decomposition of $\text{inf}(T)$ as in Proposition 11. We have that $\text{inf}(T) \models^{FO} \varphi$ iff there exists $j \in \{1, 2, \dots, n\}$, s.t., $T_i \models^{FO} \theta_i^j$ for all $i = 1, \dots, m$.

This lemma follows from Proposition 11 and another easy application of Theorem 18.

10 Composing formulas in $LTrL$

Finally we are ready to prove Lemma 3. First, we will show the lemma for finite and perpetual directed traces. Then we will show it for perpetual but not necessarily directed traces. Finally we will show it for all traces.

Lemma 21 Let $\varphi \in FO(\Sigma, I)$. Then there exists a formula $\alpha \in LTrL(\Sigma, I)$ such that for every finite or perpetual directed $T \in TR$ we have: $T \models^{FO} \varphi$ iff $T, \emptyset \models \alpha$.

Proof

Let $\varphi \in FO(\Sigma, I)$. By Lemma 6, there exists a trace consistent $\hat{\alpha} \in LTL(\Sigma)$ such that for every trace T it is the case that $T \models^{FO} \varphi$ iff $\sigma, \varepsilon \models \hat{\alpha}$ for some $\sigma \in lin(T)$. Next we set $\alpha = \|\hat{\alpha}\|$ where $\|\cdot\|$ is the map defined in Section 5 with the understanding that the formula NRC used by this map is the one constructed in Section 8.

Now suppose T is a finite or perpetual directed trace and $T \models^{FO} \varphi$. Then $\hat{\sigma}, \varepsilon \models \hat{\alpha}$ where $\hat{\sigma}$ is the normal linearization of T . We know that $\hat{\sigma}$ exists by Lemma 16. But by the property of NRC established in Lemma 17 and by Lemma 8 we know that $T, \emptyset \models \alpha$.

Now suppose that T is a finite or perpetual directed trace such that $T, \emptyset \models \alpha$. Then again by Lemma 17 and Lemma 8, we must have $\hat{\sigma}, \varepsilon \models \hat{\alpha}$ where $\hat{\sigma}$ is the normal linearization of T . But then by Lemma 6, this implies that $T \models^{FO} \varphi$. \square

We can now put together the assertions concerning the perpetual directed parts of a trace into a single assertion about the infinite (perpetual) part of the trace.

Lemma 22 Let $\varphi \in FO(\Sigma, I)$. Then there exists a formula $\alpha \in LTrL(\Sigma, I)$ such that for every perpetual trace T : $T \models^{FO} \varphi$ iff $T, \emptyset \models \alpha$.

Proof

Let us fix a $FO(\Sigma, I)$ formula φ . First, for every shape $sh = \{\Sigma_i\}_{i=1}^m$ of (Σ, I) we will construct a $LTrL$ formula α_{sh} with the property:

$$\text{for every perpetual trace } T \text{ of shape } sh: \quad T \models^{FO} \varphi \text{ iff } T, \emptyset \models \alpha_{sh} \quad (7)$$

Let us fix a shape $sh = \{\Sigma_i\}_{i=1}^m$. By Lemma 20, for the shape sh we have an array of $FO(\Sigma, I)$ formulas:

$$(\theta_1^1, \dots, \theta_m^1), (\theta_1^2, \dots, \theta_m^2), \dots, (\theta_1^n, \dots, \theta_m^n) \quad (8)$$

such that whenever a perpetual trace T is of the shape sh and $\{T_i\}_{i=1}^m$ is the decomposition of T as in Proposition 11 then:

$$\inf(T) \models^{FO} \varphi \quad \text{iff} \quad \begin{array}{l} \text{there is } j \in \{1, 2, \dots, n\} \text{ such that} \\ \text{for all } i \in \{1, 2, \dots, m\}, T_i, \emptyset \models \theta_i^j \end{array}$$

Moreover each θ_i^j is over the alphabet Σ_i .

By Lemma 21, for every θ_i^j we can find a $LTrL$ formula α_i^j such that for every perpetual directed trace T' over the alphabet Σ_i we have $T' \models^{FO} \theta_i^j$ iff $T', \emptyset \models \alpha_i^j$. Hence, for a decomposition $\{T_i = (E_i, \leq_i, \lambda_i)\}_{i=1}^m$ as above and for every $j = 1, \dots, n$ we have $T_i \models^{FO} \theta_i^j$ iff $T_i, \emptyset \models \alpha_i^j$. Now $\Sigma_i \times \Sigma_j \subseteq I$ whenever $i \neq j$ and $i, j \in \{1, 2, \dots, m\}$. Hence, we are assured that $c \subseteq E$ is a configuration of T iff $c_i = c \cap E_i$ is a configuration of T_i for each i . It is easy to establish by structural induction that for every formula γ_i over Σ_i (i.e. γ_i mentioning at most the letters in Σ_i) and for every configuration c of T , we have $T, c \models \gamma_i$ iff $T_i, c_i \models \gamma_i$. Since each α_i^j is over the alphabet Σ_i we have: $T, \emptyset \models \alpha_1^j \wedge \alpha_2^j \cdots \wedge \alpha_m^j$ iff $T_i, \emptyset \models \alpha_i^j$ for each i .

Let us denote $\alpha_1^j \wedge \cdots \wedge \alpha_m^j$ by β^j and let

$$\alpha_{sh} = \beta^1 \vee \cdots \vee \beta^n$$

It should be clear that α_{sh} satisfies the property (7). Next we observe that we can write a formula ν_{sh} in $LTrL$ such that $\inf(T), \emptyset \models \nu_{sh}$ iff $\inf(T)$ is of shape sh . Let $sh = \{\Sigma_i\}_{i=1}^m$ and

$$\Sigma_{sh} = \bigcup_{i=1}^m \Sigma_i.$$

Then

$$\nu_{sh} = \left(\bigwedge_{a \in \Sigma_{sh}} \diamond \langle a \rangle \underline{tt} \right) \wedge \left(\bigwedge_{a \notin \Sigma_{sh}} \square \sim \langle a \rangle \underline{tt} \right).$$

Clearly (Σ, I) admits only finitely many shapes. Let SH be the set of all shapes. Then consider the formula:

$$\alpha = \bigwedge_{sh \in SH} (\nu_{sh} \supset \alpha_{sh}).$$

It is easy to verify that α satisfies the property required by the lemma.

□

Finally, we show how an assertion about the finite part and an assertion about the infinite part of a trace can be put together to form an assertion about the whole trace.

Lemma 23 Let $\alpha_0, \alpha_1 \in LTrL(\Sigma, I)$. Then there exists a formula $\alpha \in LTrL(\Sigma, I)$ such that for every $T \in TR^{inf}$, $T, \emptyset \models \alpha$ iff $fin(T), \emptyset \models \alpha_0$ and $inf(T), \emptyset \models \alpha_1$.

Proof

First we define an *LTrL* formula *BORDER* that holds precisely in the configuration of T consisting of all the events in $fin(T)$:

$$\begin{aligned} \text{BORDER} = & \left(\bigwedge_{a \in \Sigma} \langle a^{-1} \rangle \underline{\text{tt}} \supset \square \sim \langle a \rangle \underline{\text{tt}} \right) \wedge \\ & \bigwedge_{b \in \Sigma} \square (\langle b \rangle \underline{\text{tt}} \supset \langle b \rangle \diamond \langle b \rangle \underline{\text{tt}}). \end{aligned}$$

Next we define *FIN* to be the formula $\diamond \text{BORDER}$. We have that for every trace T : $T, c \models FIN$ iff $c \subseteq fin(T)$ and $T, c \models BORDER$ iff $c = E_{fin}$; where E_{fin} is the set of events of $fin(T)$.

Now, with each formula $\alpha \in LTrL$ we associate the formula $fin(\alpha)$ inductively as follows.

$$\begin{aligned} fin(\underline{\text{tt}}) &= \underline{\text{tt}} & fin(\sim \alpha) &= \sim fin(\alpha) \\ fin((\alpha \vee \beta)) &= fin(\alpha) \vee fin(\beta) \\ fin(\langle a \rangle \alpha) &= \langle a \rangle (FIN \wedge fin(\alpha)) \\ fin(\alpha U \beta) &= fin(\alpha) U (FIN \wedge fin(\beta)) \end{aligned}$$

Also, with each formula β we associate the formula $inf(\beta)$ given by:

$$inf(\beta) = \diamond (\text{BORDER} \wedge \beta)$$

Now, let $T = (E, \leq, \lambda) \in TR^{inf}$ and $fin(T) = (E_{fin}, \leq_{fin}, \lambda_{fin})$ and $inf(T) = (E_{inf}, \leq_{inf}, \lambda_{inf})$. It follows from the definitions that $c \subseteq E_{fin}$ is a configuration of $fin(T)$ iff c is a configuration of T . Hence using the properties of the translation map fin defined above we can establish by structural induction on α that $T, c \models fin(\alpha)$ iff $fin(T), c \models \alpha$ for each configuration c of $fin(T)$.

Next we note that $c \subseteq E_{inf}$ is a configuration of $inf(T)$ iff $E_{fin} \cup c$ is a configuration of T . Again, by using the property of the map inf , we can

show by structural induction, that

$$T, E_{fin} \cup c \models inf(\beta) \text{ iff } inf(T), c \models \beta$$

for every configuration c of $inf(T)$. It now follows at once that for every $T \in TR^{inf}$, $T, \emptyset \models fin(\alpha_0) \wedge inf(\alpha_1)$ iff $fin(T), \emptyset \models \alpha_0$ and $inf(T), \emptyset \models \alpha_1$. \square

Clearly Lemmas 19, 22, and 23 together yield Lemma 3 and we are done with the proof of Theorem 1.

It remains to show Corollary 5. Given a formula $\alpha \in LTrL(\Sigma, I)$ we translate it into an equivalent first order formula φ and then use Lemma 6 to obtain a trace consistent formula $\hat{\alpha}$ satisfied by the linearisations of traces satisfying α . For a proof in the other direction take a trace consistent formula $\hat{\alpha} \in LTL(\Sigma)$. We first translate it into an equivalent formula $\hat{\varphi} \in FO(\Sigma)$. Next we can translate $\hat{\varphi}$ into a first order formula $\varphi \in FO(\Sigma, I)$ satisfied in the traces whose linearisations are accepted by $\hat{\varphi}$. The fact that this is possible follows from [6]. It can also be derived using the compositional theorems and the fact that our normal linearisations are definable in first order logic for perpetual directed traces. Finally we can translate φ into an equivalent formula of $LTrL(\Sigma, I)$ using Theorem 1.

11 Concluding Remarks

Our aim here has been to exhibit a temporal logic over traces with a bounded number of operators which is equivalent in expressive power to the first order theory of traces. The logic, LTrL, that we have formulated and studied here is a smooth generalization of LTL in terms of both its syntax and its semantics.

As the reader may have noticed, past formulas play a crucial role in the proof of expressive completeness of LTrL. Although there are only finitely many past formulas in the logic, at present we do not know if their use can be avoided without loss of expressive power as it is the case with LTL.

Apart from its expressive completeness, LTL admits a PSPACE decision procedure. As a consequence, LTL can be used as a specification logic and the model checking problem can be solved in an automated fashion using tools such as SPIN [12].

Unfortunately, it looks unlikely at present that LTrL can be used as a specification logic. Recently, the second author has shown that the satisfiability problem for LTrL is non-elementary hard [29]. It turns out that exploiting the fact that a model can consist of two independent linear orders, one can describe in LTrL very large counters and compare their values. This in turn allows to code long computations of Turing machines.

As a result, the model checking problem for LTrL is also going to be non-elementary hard. For the record, it might be worth while to give a precise formulation of the LTrL model checking problem.

To start with, we will formulate a slightly modified version of LTrL in terms of distributed alphabets. We will also indicate how atomic propositions can be handled in this version of our logic. However, the lower bound on the complexity of the model checking problem will hold even in the absence of atomic propositions.

Fix a finite set of agents $\mathcal{P} = \{1, 2, \dots, k\}$ with i, j ranging over \mathcal{P} . A distributed alphabet over \mathcal{P} is a family $\tilde{\Sigma} = \{\Sigma_i\}_{i \in \mathcal{P}}$ where each Σ_i is finite non-empty alphabet of actions. We set $\Sigma = \bigcup_{i \in \mathcal{P}} \Sigma_i$ and call it the global alphabet associated with $\tilde{\Sigma}$. Furthermore we define $\text{loc} : \Sigma \longrightarrow 2^{\mathcal{P}}$ via:

$$\text{loc}_{\tilde{\Sigma}}(a) = \{i \mid a \in \Sigma_i\}.$$

This induces the independence relation $I_{\tilde{\Sigma}} \subseteq \Sigma \times \Sigma$ given by:

$$a I_{\tilde{\Sigma}} b \quad \text{iff} \quad \text{loc}_{\tilde{\Sigma}}(a) \cap \text{loc}_{\tilde{\Sigma}}(b) = \emptyset$$

Thus each distributed alphabet $\tilde{\Sigma}$ induces in a canonical fashion the trace alphabet $(\Sigma, I_{\tilde{\Sigma}})$.

Next we fix a family $\{AP_i\}_{i \in \mathcal{P}}$ where AP_i is a set of atomic propositions for each i and $AP_i \cap AP_j = \emptyset$ for $i \neq j$. We set $AP = \bigcup_{i \in \mathcal{P}} AP_i$ and let p range over AP . We now define the corresponding version of LTrL to be:

$$\text{LTrL}(\tilde{\Sigma}) ::= p \mid \sim \alpha \mid \alpha \vee \beta \mid \langle a \rangle \alpha \mid \langle a^{-1} \rangle \alpha \mid \alpha \mathcal{U} \beta$$

A model for this logic is a pair $M = (T, V)$ with $T = (E, \leq, \lambda)$ being a trace over $(\Sigma, I_{\tilde{\Sigma}})$ and $V : C_T \longrightarrow 2^{AP}$ being a valuation which is required to respect $I_{\tilde{\Sigma}}$ in the following sense:

$$\text{If } c \xrightarrow{a}_T c' \text{ and } i \notin \text{loc}(a) \text{ then } V(c) \cap AP_i = V(c') \cap AP_i. \quad (9)$$

The semantics for this version of LTrL is defined in the obvious way. The restriction imposed on the valuation functions associated with models will ensure that the truth value of an atomic proposition in AP_i at a configuration c will depend only on the i -events in c . More precisely, let $T = (E, \leq, \lambda)$ be a trace over $(\Sigma, I_{\tilde{\Sigma}})$. Then E_i , the set of i -events of T , is defined to be :

$$E_i = \{e \mid e \in E \text{ and } \lambda(e) \in \Sigma_i\}.$$

Now let $M = (T, V)$ be a model and $c, c' \in C_T$ such that $c \cap E_i = c' \cap E_i$. Then for each p in AP_i we are assured by our definition of a model that $M, c \models p$ iff $M, c' \models p$.

One may wish to consider the alternative where we have a (global) set of atomic propositions P and define a model to be a pair $M = (T, V)$ with V being a valuation function which assigns to each configuration of T a subset of P . Again the semantics of the corresponding version of LTrL would be defined in the obvious way. In this case however, for all non-trivial trace alphabets (in which the independence relation is non-empty) one can easily show that the satisfiability problem is undecidable [15]. This is the reason why we require our atomic propositions and valuation functions to have a “local flavour”. From a pragmatic standpoint this is acceptable since atomic assertions concerning distributed programs are invariably of a local kind.

We are now ready to formulate the model checking problem. As the first step we define a distributed transition system over $\tilde{\Sigma} = \{\Sigma_i\}_{i \in \mathcal{P}}$ to be a family $\tilde{TS} = \{TS_i\}_{i \in \mathcal{P}}$ where for each i , we have $TS_i = (Q_i, \longrightarrow_i, q_{in}^i)$ with Q_i a set of i -local states, $\longrightarrow_i \subseteq Q_i \times \Sigma_i \times Q_i$ the i -local transition relation and $q_{in}^i \in Q_{in}$ the i -local initial state.

\tilde{TS} induces the global transition system $TS = (Q, \longrightarrow, q_{in})$ where $Q = Q_1 \times Q_2 \times \dots \times Q_k$ ($\mathcal{P} = \{1, 2, \dots, k\}$) and $q_{in} = (q_{in}^1, \dots, q_{in}^k)$ and $\longrightarrow \subseteq Q \times \Sigma \times Q$ ($\Sigma = \bigcup_{i \in \mathcal{P}} \Sigma_i$) is given by:

$$q \xrightarrow{a} q' \text{ iff } q(i) \xrightarrow{a}_i q'(i) \text{ for each } i \in \text{loc}(a) \text{ and } q(i) = q'(i) \text{ for each } i \notin \text{loc}(a)$$

We will say that \tilde{TS} is finite if TS is a finite transition system.

A distributed program over $\tilde{\Sigma}$ is a family $\tilde{Pr} = \{(TS_i, V_i)\}_{i \in \mathcal{P}}$ where $\{TS_i\}_{i \in \mathcal{P}}$ is a family of finite transition systems as above and $V_i : Q_i \longrightarrow 2^{AP_i}$.

Now let $M = (T, V)$ be a model. Then M is in the (linear time trace) behaviour of \widetilde{Pr} iff there exists a map $\rho : C_T \rightarrow Q$ such that the following conditions are satisfied:

- $\rho(\emptyset) = q_{in}$.
- If $c \xrightarrow{a}_T c'$ then $\rho(c) \xrightarrow{a} \rho(c')$ in \widetilde{TS} where $\widetilde{TS} = \{TS_i\}_{i \in \mathcal{P}}$.
- If $\rho(c) = q$ then for each i , $V(c) \cap AP_i = V_i(q(i))$.

Let $Bh(\widetilde{Pr})$ denote the set of behaviours of \widetilde{Pr} defined as above. It should be clear that every $(T, V) \in Bh(\widetilde{Pr})$ satisfies condition (9) of being a model. We will say that \widetilde{Pr} meets the specification α – denoted $\widetilde{Pr} \models \alpha$ – iff for all $M \in Bh(\widetilde{Pr})$ we have $M \models \alpha$. The model checking problem is to determine, given \widetilde{Pr} and α , whether or not $\widetilde{Pr} \models \alpha$.

It is not difficult to see that the satisfiability problem is a special case of the model checking problem. For our fixed distributed alphabet $\widetilde{\Sigma} = \{\Sigma_i\}_{i \in \mathcal{P}}$ we can define a program \widetilde{Pr} whose behaviour is the set of all possible traces over $\widetilde{\Sigma}$. Then the question whether a formula α of LTrL over $\widetilde{\Sigma}$ is satisfiable reduces to the problem of whether $\widetilde{Pr} \models \sim \alpha$. Hence the complexity of the model checking problem is also not elementary. The problem is decidable because it can be formulated in monadic second order logic of traces.

At present it is not clear whether one can identify a nice variant of LTrL which while being expressively complete will also admit an elementary time decision procedure.

References

- [1] R. Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *LICS '95*, pages 90–100, 1995.
- [2] B. Courcelle. Private communication.
- [3] V. Diekert and G. Rozenberg. *The book of traces*. World Scientific, Singapore, 1995.
- [4] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.
- [5] W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken*. PhD thesis, Institut für Informatik, Universität Stuttgart, 1994.
- [6] W. Ebinger and A. Muscholl. Logical definability on infinite traces. In *ICALP '93*, volume 700, pages 335–346, 1993.

- [7] S. Feferman and R. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.
- [8] A. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *7th Ann. ACM Symp. on Principles of Programming Languages*, pages 163–173, 1980.
- [9] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic : Mathematical Foundations and Computational Aspects*, volume 1. Clarendon Press, Oxford, G.B., 1994.
- [10] P. Gastin and A. Petit. Asynchronous cellular automata for infinite traces. In *ICALP '92*, volume 623 of *LNCS*, pages 583–594, 1992.
- [11] P. Godefroid. *Partial-order methods for the verification of concurrent systems*, volume 1032 of *LNCS*. Springer-Verlag, 1996.
- [12] G. J. Holzmann. *Design and Validation of Computer Protocols*. Software Series. Prentice Hall, 1992.
- [13] H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, 1968.
- [14] S. Katz and D. Peled. Interleaving set temporal logic. *Theoretical Computer Science* 75, pages 21-43, 1987.
- [15] K.Lodaya, R. Parikh, R.Ramanujam, and P.S.Thiagarajan. A logical study of distributed transition systems. *Information and Computation*, 119(1):91–118, 1995.
- [16] A. Mazurkiewicz. Concurrent program schemes and their interpretations. *Report DAIMI-PB-78*, Computer Science Department, Århus University, Århus, Denmark, 1978.
- [17] M. Mukund and P.S. Thiagarajan. Linear time temporal logics over traces. In *MFCS'96*, volume 1113 of *LNCS*, pages 62–92, 1996.
- [18] P. Niebert. A ν -calculus with local views for sequential agents. In *MFCS '95*, volume 969 of *LNCS*, pages 563–573, 1995.
- [19] P. Niebert. *A temporal logic for the specification and verification of distributed behaviour*. Ph.D. thesis, University of Hildesheim, Hildesheim, Germany, 1997.
- [20] D. Peled. Partial order reduction : model checking using representatives. In *MFCS'9*, volume 1113 of *LNCS*, pages 93–112, 1996.
- [21] D. Peled and A. Pnueli. Proving partial order properties. *Theoretical Computer Science* 126, pages 143-182, 1994.
- [22] D. Peled, T. Wilke and P. Wolper. An Algorithmic Approach for Checking Closure Properties of ω -Regular Languages. In *CONCUR'96*, volume 1119 of *LNCS*, Springer-Verlag, (1996) 596-610.
- [23] A. Pnueli. The temporal logic of programs. In *18th Symposium on Foundations of Computer Science*, pages 46–57, 1977.

- [24] R. Ramanujam. Locally linear time temporal logic. In *LICS '96*, pages 118–128, 1996.
- [25] P. S. Thiagarajan. A trace based extension of linear time temporal logic. In *LICS '94*, pages 438–447, 1994.
- [26] P. S. Thiagarajan. A trace consistent subset of *PTL*. In *CONCUR '95*, volume 962 of *LNCS*, Springer-Verlag, (1995) 438-452.
- [27] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Vol.B*, pages 133–192. Elsevier, 1990.
- [28] W. Thomas. On logical definability of trace languages. In V. Diekert, editor, *Workshop of the ESPRIT Basic Research Action No: 3166*, volume Report TUM-19002, Technical University of Munich,, pages 172–182, 1990.
- [29] I. Walukiewicz. Difficult configurations – on the complexity of LTrL. In *ICALP '98*, volume 1443 of *LNCS*, 1998.
- [30] G. Winskel and M. Nielsen. Models for concurrency. In S. Abramsky and D. Gabbay, editors, *Handbook of logic in computer science, Vol.3*, Oxford University Press, 1994.
- [31] A. Valmari. A stubborn attack on state explosion. *Formal Methods in System Design*, 1:297–322, 1992.
- [32] L. Zuck. *Past temporal logic*. PhD thesis, Weizmann Institute of Science, Israel, 1986.