A logical study of distributed transition systems

Kamal Lodaya[1], Rohit Parikh[2],
R. Ramanujam[1], P.S. Thiagarajan[3]

Abstract

We extend labelled transition systems to *distributed transition systems* by labelling the transition relation with a finite set of actions, representing the fact that the actions occur as a *concurrent step*. We design an action-based temporal logic in which one can explicitly talk about steps. The logic is studied to establish a variety of positive and negative results in terms of axiomatizability and decidability.

Our positive results show that the step notion is amenable to logical treatment via standard techniques. They also help us to obtain a logical characterization of two well known models for distributed systems: labelled elementary net systems and labelled prime event structures.

Our negative results show that demanding deterministic structures when dealing with a "non-interleaved" notion of transitions is, from a logical standpoint, very expressive. They also show that another well known model of distributed systems called asynchronous transition systems exhibits a surprising amount of expressive power in a natural logical setting.

[1]The Institute of Mathematical Sciences, Madras 600 113, INDIA

[2]City University of New York, New York NY 10036-8099, USA

[3]School of Mathematics, SPIC Science Foundation, 92 G.N. Chetty Road, T. Nagar, Madras 600 017 INDIA

0 Introduction

Transition systems are a simple and unifying model for representing the behaviour of distributed systems. They are used to provide the operational semantics of various process algebras such as CCS [Mil]. A number of other models of distributed systems such as elementary net systems [Thi], prime event structures [Win] and Petri nets [Rei] also have transition systems asssociated with them in a natural way to explain their operational behaviours. Consequently, a variety of logics that have been proposed to reason about the behaviours of distributed systems are based on models built out of transition systems [Pnu, ES, HM]. A classic and powerful example of such logics is the propositional μ-calculus [Koz].

The transition systems that are used in such applications are, however, sequential. A (labelled) transition in these transition systems is a triple (s, a, s') denoting that the system can perform the (single) action a at the state s and, as a result, enter the state s'. Thus it is the so-called interleaved behaviours of distributed systems that are represented by such transition systems.

It has been observed by various researchers [BC, DM, NRT] that concurrency can be more explicitly represented by enriching the transition relation, for instance by putting more information on the labels of transitions. One of the simplest ways of doing so is to consider transitions of the form (s, u, s') where u is a finite *set* of actions. The idea is that the set of actions in u can occur independently of each other (not necessarily simultaneously) at the state s and when they have all occurred, the resulting state is s'.

The aim of this paper is to study the logical consequences of admitting such an enriched transition relation. In order to focus attention on the notion of steps we design a "minimal" action-based temporal logic in which one can explicitly talk about steps and which is just about rich enough to make life interesting. We use "step-based" transition systems as Kripke frames to construct models for this logic. We then bring out the logical properties of the step notion by establishing a variety of positive and negative results in terms of axiomatizability and decidability.

To bring out the specific results, the rest of the paper is organized as follows. In the next section we introduce distributed transition systems which are transition systems based on concurrent steps. In the literature some other types of transition systems have also been called distributed transition systems [DM, Sta]. We first explain the conditions imposed on our distributed transition systems which ensure that the notion of a step indeed captures concurrency in a faithful fashion. We then show how two well-known models of concurrency, namely, prime event structures and elementary net systems, give rise to distributed transition systems in a natural way.

In Section 2, the logical language is introduced and its semantics is defined in terms of models whose underlying Kripke frames are distributed transition systems. We then propose a complete axiomatization of the valid formulas of this logic. The completeness proof is based on standard filtration techniques borrowed from research on PDL (Propositional Dynamic Logic) [KP]. A consequence of the completeness argument is that the satisfiability problem for this logic is decidable in nondeterministic exponential time. On the other hand, we have a deterministic exponential time lower bound.

In Section 3 we establish the somewhat surprising result that our logic cannot separate the class of models based on (distributed transitions yielded by) prime event structures from the general class of models. As a result, the axiomatization in Section 2 is complete for the restricted class as well.

Using the well-known relationships between elementary net systems and prime event structures [NPW] we then show that similar results can also be established for the subclass of models based on elementary net systems.

Starting from Section 4 we begin to study subclasses of distributed transition systems and establish a sequence of (predominantly negative) results about subclasses of distributed transition systems. Section 4 shows that the set of valid formulas over the class of deterministic models can be axiomatized in a simple (and finitary) fashion. The completeness argument is quite involved; the reason being, as we show in Section 5, that validity is not decidable. Here and in subsequent sections we make heavy use of the negative results based on domino problems due to Harel [Har85].

The results established so far are based on distributed transition systems over an infinite alphabet set. Due to the unusual mixture of modalities in our logic, there is a good deal of difference between finite and infinite alphabets. This is especially so in the presence of determinacy. This is brought out in Section 6. We show that the satisfiability problem over the class of deterministic models is Σ_1^1-hard, if we restrict the set of actions to be a finite set (but containing at least two elements!). As a result, validity over this class of models is not axiomatizable. In the next section we show that the satisfiability problem over the class of *finite* deterministic models is r.e.-hard and hence not decidable. Once again, an easy consequence is that validity over this class of models is not axiomatizable.

The proof techniques that we develop to establish our results indicate that various generalizations are possible. On the positive side, the results of Section 2 and Section 3 go through – with some additional machinery – even if we replace steps (i.e. finite sets) with finite multisets or finite pomsets [Pra86] as labels of transitions. On the negative side, it turns out the various undecidability results go through if we use, instead of deterministic distributed transition systems, transition systems based on *traces* [Maz]. These additional negative results are presented in Section 8. The lesson to be drawn here is that in the presence of concurrency (as captured by steps) it is not only determinacy but even a kind of partial commutativity property (implied by the presence of steps) which makes the logic very expressive.

In Section 9 we point out how the study that we have carried out can also be done for a natural generalization of PDL. We also sketch briefly how a more powerful modality based on the notion of steps can lead to a finitary axiomatization of validity over the general class of models. In the concluding section, we discuss related literature in more detail.

A logic for distributed transition systems was first studied in [LRT], where only a finite alphabet was considered. Theorem 6.5 was proved in that paper. The high undecidability of the logic for deterministic distributed transition systems over a finite alphabet (Theorem 6.10) was proved in [Parikh].

1 Distributed transition systems

In this section we introduce distributed transition systems which will serve as the frames for our logic. We will show how such transition systems arise in the study of two well-known models of distributed systems.

Recall that a (sequential) *transition system* is a triple $TS = (S, \Sigma, \rightarrow)$ where S is a set of states,

Σ is a set of actions and $\rightarrow \subseteq S \times \Sigma \times S$ is the (labelled) transition relation. If $(s, a, s') \in \rightarrow$ then the idea is that the action a *can* occur at state s and, as a result, the system assumes the state s'.

The essential feature of a distributed transition system is that the transition relation is generalized to (s, u, s'), where u is a finite set of actions. The actions in u are interpreted as a *concurrent step*. This means that further conditions have to be placed on the transition relation.

We will use the notation $\wp(X)$ for the powerset of a set X and $\wp_{fin}(X)$ for the set of finite subsets of X.

Definition 1.1 *A* **distributed transition system (dts)** *is a triple* $DTS = (S, \Sigma, \rightarrow)$ *where*

1. S *is a set of* states
2. Σ *is a set of* actions
3. $\rightarrow \subseteq S \times \wp_{fin}(\Sigma) \times S$ *is the* step transition relation *satisfying for all* s, s' *in* S:
   (a) $s \xrightarrow{\emptyset} s'$ *iff* $s = s'$.
   (b) *for all* $u \in \wp_{fin}(\Sigma)$, *if* $s \xrightarrow{u} s'$ *then there exists a function* $f : \wp(u) \rightarrow S$ *such that* $f(\emptyset) = s$, $f(u) = s'$ *and for every* $v_1 \subseteq v_2 \subseteq u$ *such that* $v_2 - v_1 \neq u$, *it is the case that* $f(v_1) \xrightarrow{v_2 - v_1} f(v_2)$.

The function f is said to be a *u-cube*. We will let $\mathcal{F}[u, X]$ denote the set of functions from $\wp(u)$ into the set X. As in the above definition, we will often write $s \xrightarrow{u} s'$ instead of $(s, u, s') \in \rightarrow$. The letters u, v with or without subscripts will range over $\wp_{fin}(\Sigma)$. For $a \in \Sigma$, we will also write $\xrightarrow{a}$ instead of $\xrightarrow{\{a\}}$.

Figure 1 is a graphical representation of an $\{a, b, c\}$-cube. The nodes of the graph represent the states of the system. The edges, labelled by actions from Σ, reflect the transition relation $\rightarrow$. To avoid cluttering up the pictures, when we show a concurrent step, we do not display all the smaller substeps but only the actions (steps of size 1). This convention is followed in Figure 1 and all subsequent figures. Where Σ is clear from the context, we will often display a dts as just an ordered pair $(S, \rightarrow)$ and call it a dts *over* Σ.

Suppose $s \xrightarrow{u} s'$ in a dts. Then the idea is that the actions in u can occur at s with no order over their occurrences; and when they have all occurred, the resulting state is s'. We say that the *step* u is enabled at s. The existence of the u-cube guarantees that this mutual independence of the actions in u at s holds at all the states reached through a part of the step for the "residual" substeps.

It is important to note that clause (3.b) in the definition of a dts is merely an implication. The existence of a u-cube does not guarantee the existence of a concurrent step. Figure 2 shows all the actions required for an $\{a, b\}$ step, but there is no concurrent step in the picture. All the inductively smaller substeps of a concurrent u-step may exist, but this may be accidental. It is the u arrow that shows that the substeps form part of a concurrent step.

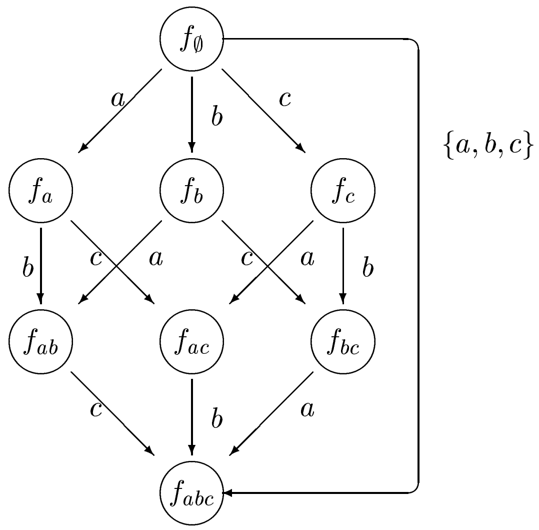


Figure 1: An $\{a, b, c\}$-cube

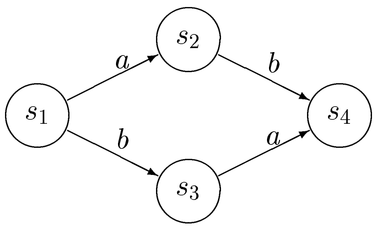


Figure 2: A dts without any concurrent step

This is a characteristic of partial order models of distributed systems in which concurrency is not identified with nondeterministic interleaving. In fact it is possible at the same state to have a concurrent step as well as an interleaving of the step performed but leading to two different states. This is illustrated in Figure 3. We will show later how this arises in some typical partial order models of concurrency.

Further, the u-cube guarantees more than the fact that a step can be broken up into all possible substeps. To bring this out let us consider replacing clause (3.b) in the definition by:

- For all s, s' in S, for all $u \in \wp_{fin}(\Sigma)$, if $s \xrightarrow{u} s'$ then there exists a function f in $\mathcal{F}[u, S]$ such that $f(\emptyset) = s$ and $f(u) = s'$ and for every $v \subseteq u$, it is the case that $s \xrightarrow{v} f(v) \xrightarrow{u-v} s'$.

For the transition system in Figure 4, we can consistently have an $\{a, b, c\}$ step between $s_\emptyset$ and s_{abc} if we accept this weaker condition (and fill in all the intermediate substeps), but there is no $\{a, b, c\}$-cube f in the sense of Definition 1.1, since $f(b)$ cannot be assigned a suitable value.

An important notion in transition systems is that of reachability. Given the transition system $TS = (S, \Sigma, \rightarrow)$, we define the **reachability set** of $s_0 \in S$, denoted $\mathcal{R}_{TS}(s_0)$, as the least subset of S containing s_0 satisfying:

$$\text{If } s \in \mathcal{R}_{TS}(s_0), a \in \Sigma \text{ and } s \xrightarrow{a} s', \text{ then } s' \in \mathcal{R}_{TS}(s_0).$$

We write $\mathcal{R}(s_0)$ if the underlying transition system is clear from the context.

$TS = (S, \rightarrow, s_0)$ is said to be a **pointed** dts if $(S, \rightarrow)$ is a dts, $s_0 \in S$ and $S = \mathcal{R}_{TS}(s_0)$.

In this paper, we only consider *countable* dts's, that is, in $TS = (S, \Sigma, \rightarrow)$, S, Σ and $\rightarrow$ are all at most countable.

We have chosen the strong definition of dts's after examining a number of partial order models of distributed systems. We will consider two such models: event structures and net systems. Our presentation will be brief and the interested reader is referred to [Win, Thi, NRT] for more background material.

A **prime event structure** is a triple ES $= (E, \leq, \#)$ where

- E is a set of event occurrences
- $\leq \subseteq E \times E$ is a partial ordering relation called the *causality* relation.
- $\# \subseteq E \times E$ is an irreflexive and symmetric relation called the *conflict* relation.
- $\#$ is inherited via $\leq$ in the sense that $e_1 \# e_2$ and $e_2 \leq e_3$ imply $e_1 \# e_3$ for every e_1, e_2, e_3 in E.

Figure 5 is an example of an event structure. The squiggly lines represent the "minimal" elements of the $\#$ relation. The causality relation is shown in the form of the associated Hasse diagram. The $\#$ relation is then uniquely determined by the last part of the definition above. In this event structure, $e_1 \# e_6$ because $e_1 \# e_2 \leq e_6$.

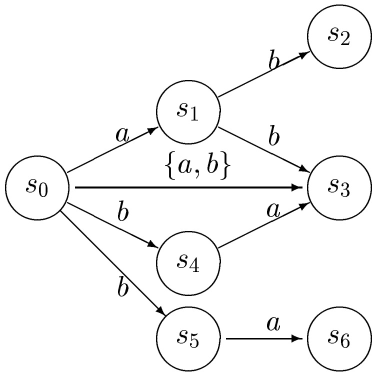


Figure 3: A nondeterministic dts

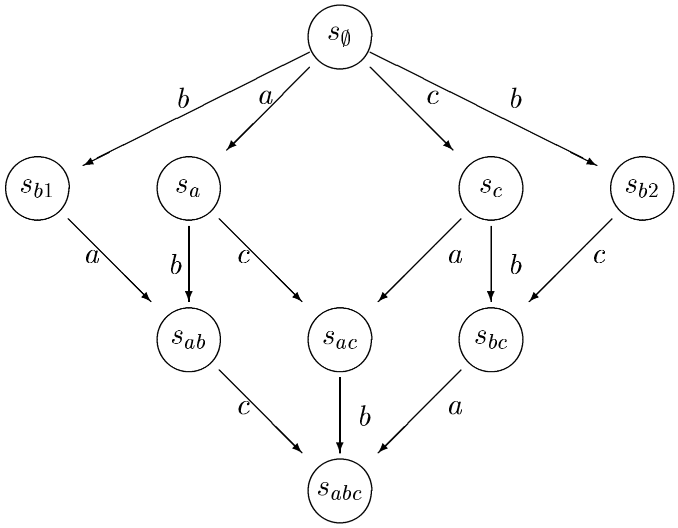


Figure 4: A non-cube

Events which are not ordered by $\leq$ and not in conflict are interpreted as being concurrent. Formally for the event structure $ES = (E, \leq, \#)$, we define $co_{ES} \subseteq E \times E$ as:

$$e_1 \; co_{ES} \; e_2 \text{ iff not } (e_1 \leq e_2 \text{ or } e_2 \leq e_1 \text{ or } e_1 \# e_2)$$

We will drop the subscript if the event structure ES is understood from the context. For example, in Figure 5, $e_6 \; co \; e_7$.

An event structure is said to be **finitary** if every event has at most a finite number of events causing it. For formalizing this idea and for defining the notion of a state it will be convenient to adopt the following notation.

Let $ES = (E, \leq, \#)$ be an event structure and $X \subseteq E$. Then $\downarrow X$ is defined to be

$$\{e' \mid \exists e \in X : e' \leq e\}.$$

In case $X = \{e\}$ we will write $\downarrow e$ instead of $\downarrow\{e\}$. Now ES is finitary iff $\downarrow e$ is finite for every e in E. In this paper we consider only finitary event structures.

For an event to occur in a computation all the events that cause it must have occurred. No two events that are in conflict can both occur in a computation. These considerations lead to the following notion of "state" for an event structure.

Let $ES = (E, \leq, \#)$ be an event structure. Then $x \subseteq E$ is a **configuration** iff

(i) $x = \downarrow x$ (downward closed)
(ii) $(x \times x) \cap \# = \emptyset$ (conflict-free)

Let C_{ES} denote the set of *finite* configurations of the event structure $ES = (E, \leq, \#)$. The occurrence of an event or a finite set of concurrent events causes the configuration to change, in effect giving a transition system. Define now the step transition relation $\rightarrow_{ES} \subseteq C_{ES} \times \wp_{fin}(E) \times C_{ES}$ of ES (over E) as

$$x \xrightarrow{u} x' \text{ iff } x' = x \cup u, \; x \cap u = \emptyset \text{ and } \forall v \subseteq u : x \cup v \text{ is a configuration}$$

Proposition 1.2 *$(C_{ES}, E, \rightarrow_{ES})$ is a dts.*

Proof: It suffices to verify that $(C_{ES}, E, \rightarrow_{ES})$ satisfies the step condition. For convenience, we will write $\rightarrow_{ES}$ as $\rightarrow$ through the rest of the proof.

Clearly $x \xrightarrow{\emptyset} x'$ iff $x = x'$ for every x, x' in C_{ES}. So assume that $u \neq \emptyset$ and $x \xrightarrow{u} x'$. Define $f \in \mathcal{F}[u, C_{ES}]$ by:

$$f(v) = x \cup v, \text{ for all } v \subseteq u.$$

Clearly f is well-defined and $f(\emptyset) = x$ and $f(u) = x'$. It is easy to verify that f is in fact a u-cube. $\square$

Note that the dts produced by the event structure is *deteministic.* In applications, one often works with labelled event structures. We can also associate with the labelled event structure a dts over the label set. This observation will help establish one of the results of this paper (Section 3). Let Σ be a set of labels. A **Σ-labelled event structure** is a quadruple $ES = (E, \leq, \#, \phi)$ where

- $(E, \leq, \#)$ is an event structure called the underlying event structure of ES.
- $\phi : E \to \Sigma$ is the *labelling function.*

The labelling function ϕ can be extended pointwise to finite subsets of E.

We assume that the notions we have so far developed for event structures are transported to labelled event structures via their underlying event structures in the obvious way. A slight hitch is that in associating a dts over Σ with a Σ-labelled event structure, concurrent steps have to be defined using *multisets* rather than sets. We would like to stick to the simpler notation of sets, which we do in this paper, using "concurrency preserving" labelling functions. However, our results do not depend upon this and can be generalized if required.

Let $ES = (E, \leq, \#, \phi)$ be a Σ-labelled event structure. Then ϕ is said to be **concurrency preserving** in case $e_1 \; co_{ES} \; e_2$ implies $\phi(e_1) \neq \phi(e_2)$ for every e_1, e_2 in E. Suppose ES is labelled preserving concurrency. Let

$$TS_{ES} = (C_{ES}, \Sigma, \Rightarrow_{ES}) \text{ where } \Rightarrow_{ES} \stackrel{\text{def}}{=} \{(x, \phi(u), x') \mid (x, u, x') \in \to_{ES}\}.$$

Proposition 1.3 *TS_{ES} is a dts over Σ.*

Proof: Follows easily from the fact that $(C_{ES}, \to_{ES})$ is a dts over E. □

Henceforth, by a "labelled event structure" we shall mean a finitary event structure with a concurrency preserving labelling function.

Next we wish to show that elementary net systems which are a basic model in net theory also give rise to dts's.

An **elementary net system** is a tuple $\mathcal{N} = (B, E, F, c_{in})$ where

- $N_{\mathcal{N}} = (B, E, F)$ is called the underlying *net* of $\mathcal{N}$. B is a set of *conditions* and E a set of *events* (disjoint from B). The *flow* relation $F \subseteq (B \times E) \cup (E \times B)$ satisfies:

$$\forall x \in B \cup E : \exists y \in E \cup B : (x, y) \in F \text{ or } (y, x) \in F.$$

- $c_{in} \subseteq B$ is the *initial case.*

For e in E we let ${}^\bullet e$ denote the set of pre-conditions and $e^\bullet$ the set of post-conditions of e, defined as:

$${}^\bullet e \stackrel{\text{def}}{=} \{b \mid (b, e) \in F\}$$

$$e^\bullet \stackrel{\text{def}}{=} \{b \mid (e, b) \in F\}.$$

For a subset of events $X \subseteq E$, ${}^\bullet X$ and $X^\bullet$ are defined by taking the pointwise union.

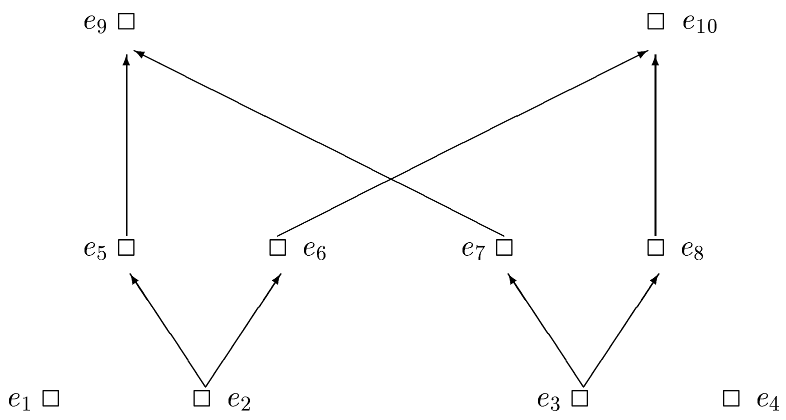


Figure 5: An event structure

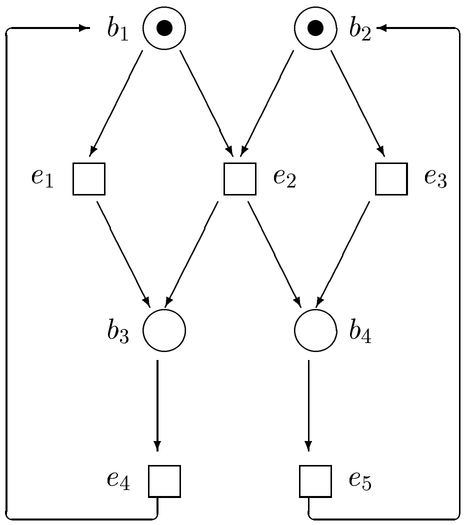


Figure 6: An elementary net system

Figure 6 is an example of an elementary net system. We have used the conventional graphical notation for nets — conditions are represented by circles, events by boxes and the flow relation by directed arcs. The "marked" conditions denote the initial case c_{in}.

A state of a net system – usually called a *case* – consists of a subset of the conditions holding concurrently. An event can occur at a case iff all its pre-conditions hold and none of its post-conditions do at the case. When an event occurs all its pre-conditions cease to hold and all its post-conditions begin to hold. A step of events can occur at a case – as a concurrent step – if each of them can occur individually and their F-neighbourhoods are pairwise disjoint. These ideas can be formalized as follows:

Let $N = (B, E, F)$ be a net. Then $Ind_N \subseteq E \times E$ is given by:

$$e_1 \ Ind_N \ e_2 \text{ iff } ({}^\bullet e_1 \cup e_1{}^\bullet) \cap ({}^\bullet e_2 \cup e_2{}^\bullet) = \emptyset.$$

The step transition relation $\rightarrow_N \subseteq \wp(B) \times \wp_{fin}(E) \times \wp(B)$ is given by:

$$c \xrightarrow{u} c' \text{ iff } c - c' = {}^\bullet u, \ c' - c = u^\bullet \text{ and } \forall e_1, e_2 \in u : e_1 = e_2 \text{ or } e_1 \ Ind_N \ e_2$$

Note that if $c \subseteq B$ and ${}^\bullet e \subseteq c$ but $e^\bullet \cap c \neq \emptyset$, then e is *not* enabled at c.

Let $\mathcal{N} = (B, E, F, c_{in})$ be an elementary net system and $\rightarrow_N$ the step transition relation of the underlying net $N = (B, E, F)$. Then $\mathcal{C}_\mathcal{N}$ is the state space of $\mathcal{N}$ and it is the least subset of $\wp(B)$ containing c_{in} and satisfying:

$$\text{If } c \in \mathcal{C}_\mathcal{N} \text{ and } (c, u, c') \in \rightarrow_N \text{ then } c' \in \mathcal{C}_\mathcal{N}.$$

Let $\rightarrow_\mathcal{N}$ be $\rightarrow_N$ restricted to $\mathcal{C}_\mathcal{N} \times \wp_{fin}(E) \times \mathcal{C}_\mathcal{N}$.

Proposition 1.4 *$(\mathcal{C}_\mathcal{N}, E, \rightarrow_\mathcal{N})$ is a dts over E.*

Proof: Suppose $(c, u, c') \in \rightarrow_\mathcal{N}$. Define $f \in \mathcal{F}[u, \mathcal{C}_\mathcal{N}]$ by $f(v) = (c \cup v^\bullet) - {}^\bullet v$, for every $v \subseteq u$. Now it is easy to verify that Definition 1.1 is satisfied. □

As in the case of event structures, the dts associated with a net system is deterministic, and it will be useful to consider labelled net systems. A **Σ-labelled elementary net system** $\mathcal{N} = (B, E, F, c_{in}, \phi)$ is defined analogously to a labelled event structure. Our labelling function will be required to preserve concurrency.

Let $\mathcal{N} = (B, E, F, c_{in})$ be a net system. Then $co_\mathcal{N} \subseteq E \times E$ is given by:

$$co_\mathcal{N} = \{(e_1, e_2) \mid e_1 \neq e_2 \text{ and } \exists c, c' \in \mathcal{C}_\mathcal{N} : (c, \{e_1, e_2\}, c') \in \rightarrow_\mathcal{N}\}.$$

Notice that $co_\mathcal{N} \subseteq Ind_{N_\mathcal{N}}$ and in general this inclusion is proper.

We can now define the structure

$$TS_\mathcal{N} = (\mathcal{C}_\mathcal{N}, \Sigma, \Rightarrow_\mathcal{N}) \text{ where } \Rightarrow_\mathcal{N} = \{(c, \phi(u), c') \mid (c, u, c') \in \rightarrow_\mathcal{N}\}.$$

Proposition 1.5 *$TS_\mathcal{N}$ is a dts over Σ.*

Thus elementary net systems also lead to dts's in a natural way.

2 A logic for distributed transition systems

In this section we introduce the logic which will be the focal point of our study. With distributed transition systems playing the role of Kripke frames, we first develop the semantics of the language. We then provide a complete axiomatization of the set of valid formulas and show that the logic is decidable.

Fix a countably infinite set of atomic propositions $P = \{p_0, p_1, \ldots\}$ and a countably infinite alphabet of atomic actions Σ. The formulas of our language Step-TL (temporal logic with concurrent steps) are specified inductively as:

- Every member of P is a formula.
- If α and β are formulas then so are $\sim\alpha, \alpha \vee \beta, \Diamond\alpha$ and $\langle u\rangle\alpha$, for $u \in \wp_{fin}(\Sigma)$.

We let $\alpha, \beta, \gamma, \delta$ with or without subscripts range over formulas. When u is a singleton, say $u = \{a\}$, we write $\langle a\rangle\alpha$ instead of $\langle\{a\}\rangle\alpha$.

A **model** is a pair $M = (TS, V)$ where $TS = (S, \rightarrow)$ is a dts over Σ and $V : S \rightarrow \wp(P)$ is a **valuation function**.

Let $M = ((S, \rightarrow), V)$ be a model, $s \in S$. Then the notion of α holding at the state s in the model is denoted by $M, s \models \alpha$ and is defined inductively as follows:

- $M, s \models p$ iff $p \in V(s)$.
- $M, s \models \sim\alpha$ iff $M, s \not\models \alpha$.
- $M, s \models \alpha \vee \beta$ iff $M, s \models \alpha$ or $M, s \models \beta$.
- $M, s \models \Diamond\alpha$ iff there exists $s' \in \mathcal{R}(s)$ such that $M, s' \models \alpha$.
- $M, s \models \langle u\rangle\alpha$ iff there exists s' such that $s \xrightarrow{u} s'$ and $M, s' \models \alpha$.

The derived connectives of propositional calculus such as $\wedge$, $\supset$ and $\equiv$ are defined in terms of $\sim$ and $\vee$ in the usual way. We let $True$ stand for the formula $p_0 \vee \sim p_0$ and $False$ for $\sim True$.

The derived modalities $\Box$ and $[u]$ are given by:

$$\Box\alpha \stackrel{\text{def}}{=} \sim\Diamond\sim\alpha$$

$$[u]\alpha \stackrel{\text{def}}{=} \sim\langle u\rangle\sim\alpha$$

It can be easily checked that

$$M, s \models \Box\alpha \text{ iff for every } s' \in \mathcal{R}(s), M, s' \models \alpha$$

$$M, s \models [u]\alpha \text{ iff for every } s' \text{ such that } s \xrightarrow{u} s' : M, s' \models \alpha$$

The formula α is **satisfiable** if $M, s \models \alpha$ in some model $M = ((S, \rightarrow), V)$ with $s \in S$. α is valid in the model M if $M, s \models \alpha$ for every $s \in S$. α is **valid** (denoted $\models \alpha$) if α is valid in every model M.

Step-TL can be used to express a variety of properties concerning the occurrence patterns of actions in a dts. A typical safety property would be $\Box[\{a_1, a_2\}]False$ stating that at no reachable (global) state can the actions a_1 and a_2 occur concurrently. Clearly liveness properties can also be stated in the usual fashion. For example if the dts models an elementary net system then $\Box\Diamond\langle a\rangle True$ expresses the fact that from every reachable case (state) it is possible to obtain a case at which the action a is enabled.

We now propose an axiomatization of the set of valid formulas.

Axiom System ND

Axiom schemes

$(A0)$	All the substitutional instances of the tautologies of PC
$(A1)$	$\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$
$(A2)$	$\Box\alpha \supset [u]\alpha \wedge \Box\Box\alpha$
$(A3)$	$[u](\alpha \supset \beta) \supset ([u]\alpha \supset [u]\beta)$
$(A4)$	$\alpha \equiv \langle\emptyset\rangle\alpha$

Inference rules

$(MP)\ \dfrac{\alpha, \quad \alpha \supset \beta}{\beta} \qquad (TG)\ \dfrac{\alpha}{\Box\alpha}$

Let $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$ and $\alpha \in \Gamma$.

$$(Step)\ \dfrac{\gamma_1 \vee \ldots \vee \gamma_k}{\langle u\rangle\alpha \supset \bigvee\limits_{\substack{f \in \mathcal{F}[u,\Gamma], \\ f(u)=\alpha}} \Big(\bigwedge\limits_{v \subseteq u} \langle v\rangle \bigwedge\limits_{v \subseteq v' \subseteq u} \langle v' - v\rangle f(v')\Big)}$$

As usual, by a *thesis* we will mean a formula α which is derivable in a finite number of steps from the axioms using the inference rules. This is denoted by $\vdash \alpha$. α is said to be *consistent* if $\sim\alpha$ is not a thesis. The finite set of formulas $\{\alpha_1, \ldots, \alpha_n\}$ is consistent if their conjunction $\alpha_1 \wedge \ldots \wedge \alpha_n$ is. A set of formulas is consistent if every finite subset of it is.

Most of our axiom schemes and inference rules are standard or easy adaptations of standard ones [Krö, Pnu]. Characteristic of our system are the axiom scheme $(A4)$ and the inference rule $(Step)$. The former specifies that each state can be reached from itself through the empty step. Moreover the empty step performed at a state leads back to the same state.

Note that $(Step)$ represents a finite presentation of an infinite set of inference rules: one for each set of formulas Γ. In essence $(Step)$ says that if $\langle u\rangle\alpha$ holds at a state s in the model M, then there exists a state s' such that α holds at s' and there is a u-cube from s to s'. (Think of Γ as a finite set of "descriptions" of states of a dts.)

Consider the following simple way of stating this:

$$\langle u\rangle\alpha \supset \bigwedge_{v \subseteq u} \langle v\rangle\langle u - v\rangle\alpha$$

This formula would be a thesis in our system. It merely states that a step can be arbitrarily broken up into substeps. However there is more to the semantics of the $\overset{u}{\to}$ relation, as we observed in Section 1. It demands the existence of a function which fixes once and for all the "state of affairs" that might prevail at the intermediate states occurring in the u-cube. Specifically, if each of the states in the u-cube satisfy one among a set of properties then the function must fix a specific property for each intermediate state in the u-cube. In particular note that, due to nondeterminism, for each $v \subseteq u$ there might be several v-successors at s and the function must determine which belongs to the u-cube in question.

We can use a finite set of inference rules for our axiomatization, but then the inference rule $(Step)$ is replaced by an infinite set of axiom schemes [LRT]. Let $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$ and $\alpha \in \Gamma$.

$$(AStep) \quad \langle u\rangle\alpha \wedge \bigwedge_{v\subseteq u}[v](\gamma_1 \vee \ldots \vee \gamma_k) \supset \bigvee_{\substack{f \in \mathcal{F}[u,\Gamma],\\ f(u)=\alpha}} (\bigwedge_{v\subseteq u}\langle v\rangle \bigwedge_{v\subseteq v'\subseteq u}\langle v'-v\rangle f(v'))$$

Observe that the step axioms and rules have a consequent that is double exponential in the size of the antecedent.

We do not know whether it is possible to axiomatize this logic with a finite set of inference rules and a finite set of axiom schemes. However, we *can* so axiomatize a logic in a more expressive language; see Section 9.2.

Note that the axiom system in itself does not force models to be distributed transition systems – tree models (with action-labelled edges) could suffice, and the axioms would then provide closure conditions on the models. However, we are primarily interested in dts's which arise naturally in concurrency theory and we study this logic in an attempt to characterize dts's (in the sense of standard modal logic [HC]).

Theorem 2.1 (Soundness) *If* $\vdash \alpha$ *then* $\models \alpha$.

Proof: We will only verify the soundness of the inference rule $(Step)$. The soundness of the axioms and the other inference rules is easy to check.

So suppose the disjunction of a set Γ of formulas $\gamma_1, \ldots, \gamma_k$ is valid, $\alpha \in \Gamma$. Let $M = ((S, \to), V)$ be a model and $s \in S$ such that $M, s \models \langle u\rangle\alpha$.

Hence there is an s' with α holding and a u-cube g from s to it. We define the required function $f \in \mathcal{F}[u, \Gamma]$ as follows.

Suppose $v \subset u$. $M, g(v) \models \gamma_1 \vee \ldots \vee \gamma_k$. Hence some formula in Γ must be satisfied at $g(v)$. To be specific, let $f(v)$ be the formula γ_j where j is the least index in $\{1, \ldots, k\}$ such that $M, g(v) \models \gamma_j$. Finally, let $f(u) = \alpha$.

The soundness of the axiom now follows easily from $s \overset{v}{\to} g(v)$ and $g(v) \xrightarrow{v'-v} g(v')$ for every $v \subseteq v' \subseteq u$. □

The following theses and derived inference rules will be required for proving completeness.

Theses

$(T1)$ $[u]\alpha \wedge \langle u\rangle\beta \supset \langle u\rangle(\alpha \wedge \beta)$
$(T2)$ $\Box\alpha \wedge \Diamond\beta \supset \Diamond(\alpha \wedge \beta)$
$(T3)$ $[u](\alpha \wedge \beta) \equiv [u]\alpha \wedge [u]\beta$
$(T4)$ $\Box(\alpha \wedge \beta) \equiv \Box\alpha \wedge \Box\beta$
$(T5)$ $\langle u\rangle(\alpha \wedge \beta) \supset \langle u\rangle\alpha \wedge \langle u\rangle\beta$
$(T6)$ $\langle u\rangle(\alpha \vee \beta) \equiv \langle u\rangle\alpha \vee \langle u\rangle\beta$
$(T7)$ $\Diamond(\alpha \vee \beta) \equiv \Diamond\alpha \vee \Diamond\beta$
$(T8)$ $\Diamond(\alpha \wedge \beta) \supset \Diamond\alpha \wedge \Diamond\beta$
$(T9)$ $\Box\alpha \supset [u]\Box\alpha$
$(T10)$ $\Box\alpha \supset \alpha$

Derived rules

$$(uG)\ \frac{\alpha}{[u]\alpha} \qquad (DR1)\ \frac{\alpha \supset \beta}{\langle u\rangle\alpha \supset \langle u\rangle\beta} \qquad (DR2)\ \frac{\alpha \supset \beta}{\Diamond\alpha \supset \Diamond\beta}$$

The derivations are quite easy (see for instance [Bur]) and hence we omit them.

The *closure* of a formula will play a crucial role in the completeness proof.

Definition 2.2 *Let α be a formula.*

1. *$CL'(\alpha)$ is the least set of formulas containing α which satisfies:*
   (a) *If $\sim\beta \in CL'(\alpha)$ then $\beta \in CL'(\alpha)$*
   (b) *If $\beta_1 \vee \beta_2 \in CL'(\alpha)$ then $\beta_1, \beta_2 \in CL'(\alpha)$*
   (c) *If $\langle u\rangle\beta \in CL'(\alpha)$ then $\beta \in CL'(\alpha)$*
   (d) *If $\Diamond\beta \in CL'(\alpha)$ then $\beta \in CL'(\alpha)$*
2. *$CL(\alpha)$, the* **closure** *of α, is given by:*
$$CL(\alpha) \stackrel{\text{def}}{=} CL'(\alpha) \cup \{\sim\beta \mid \beta \in CL'(\alpha)\}$$
3. *An* **atom** *generated by α is a maximal consistent subset of $CL(\alpha)$.*
4. *$AT(\alpha)$ is the set of atoms generated by α.*
5. *$Voc(\alpha)$, the (closure) vocabulary of α, is given by:*
$$Voc(\alpha) = \bigcup\{u \mid \text{there is a } \langle u\rangle\beta \in CL(\alpha)\}$$

It is easy to check that there exists a constant $c > 0$ such that if α is of length n then $CL(\alpha)$ is of size at most cn and hence $AT(\alpha)$ is of size at most 2^{cn}.

The completeness proof will consist of showing that every consistent formula is satisfiable. For the rest of the section, fix a consistent formula α_0 and an action $d \in \Sigma - Voc(\alpha_0)$. Note that d exists because $Voc(\alpha_0)$ is finite and Σ isn't.

For convenience, we will assume the parameter α_0 and write CL, AT and Voc. Clearly CL is nonempty and since α_0 is consistent, AT is nonempty. Further, each atom is nonempty since the empty set is not maximal (one can always consistently add α_0). For every atom w, we let $\widehat{w}$ denote the conjunction of the formulas contained in it. For a nonempty set $W = \{w_1, \ldots, w_k\} \subseteq AT$, $\widetilde{W}$ denotes the formula $\widehat{w_1} \vee \ldots \vee \widehat{w_k}$. The next result can be obtained by applying the machinery of propositional calculus [Krö].

Proposition 2.3 *Let $w, w' \in AT$.*

1. *If $\alpha \in w$ then $\vdash \widehat{w} \supset \alpha$*
2. *($\widehat{w} \wedge \alpha$ is consistent and $\alpha \in CL$) iff $\alpha \in w$.*
3. *$\widehat{w} \wedge \widehat{w'}$ is consistent iff $w = w'$.*
4. *$\vdash \widetilde{AT}$.*

The set AT can be used to construct a model for α_0. The underlying dts $TS_0 = (AT, \rightarrow)$ can be defined as:

$$\begin{aligned} \rightarrow &\stackrel{\text{def}}{=} \{(w, u, w') \mid \widehat{w} \wedge \langle u \rangle \widehat{w'} \text{ is consistent and } u \subseteq Voc\} \\ &\cup \{(w, \{d\}, w') \mid \widehat{w} \wedge \Diamond \widehat{w'} \text{ is consistent}\} \end{aligned}$$

Recall that $d \in \Sigma - Voc$.

Lemma 2.4 *TS_0 is a dts over $Voc \cup \{d\}$.*

Proof: $w \xrightarrow{\emptyset} w'$ iff $\widehat{w} \wedge \langle \emptyset \rangle \widehat{w'}$ is consistent iff, by axiom $(A4)$, $\widehat{w} \wedge \widehat{w'}$ is consistent. By Proposition 2.3(iii), this holds if and only if $w = w'$.

Next suppose that $w \xrightarrow{u} w'$. We must establish the existence of a u-cube from w to w' in TS_0. First note that, by the definition of $\rightarrow$, either $u \subseteq Voc$ or $u = \{d\}$. If $u = \{d\}$, the function is obvious, so suppose $u \subseteq Voc$. Then $\widehat{w} \wedge \langle u \rangle \widehat{w'}$ is consistent. Let $AT = \{w_1, \ldots, w_k\}$. Since $w' \in AT$ and $\vdash \widetilde{AT}$ by Proposition 2.3(iv), we can apply $(Step)$ to get a function f in $\mathcal{F}[u, \{\widehat{w_1}, \ldots, \widehat{w_k}\}]$ such that $f(u) = \widehat{w'}$ and the following formula is consistent:

$$\widehat{w} \wedge \bigwedge_{v \subseteq u} \langle v \rangle \bigwedge_{v \subseteq v' \subseteq u} \langle v' - v \rangle f(v')$$

Using axiom $(A4)$ and Proposition 2.3(iii), observe that $f(\emptyset)$ must be $\widehat{w}$. Consider $v \subseteq v' \subseteq u$. We have $\langle v \rangle (\langle \emptyset \rangle f(v) \wedge \langle v' - v \rangle f(v'))$ is consistent. By the derived rule (uG), the formula $\langle \emptyset \rangle f(v) \wedge \langle v' - v \rangle f(v')$ is consistent. Using $(A4)$, we see that $f(v) \wedge \langle v' - v \rangle f(v')$ is consistent.

Now define $g \in \mathcal{F}[u, AT]$ by $g(v) = w_i$ such that $\widehat{w_i} = f(v)$. It is easy to observe that g satisfies the conditions for a u-cube from w to w'. $\square$

The next intermediate result is useful in the proof of completeness.

Lemma 2.5 *Let $w, w' \in AT$ and $u \subseteq Voc \cup \{d\}$ such that $w \xrightarrow{u} w'$. If $\Diamond\alpha \in w'$ then $\Diamond\alpha \in w$.*

Proof: Suppose $w \xrightarrow{u} w'$, $w' \subseteq Voc$. Then $\widehat{w} \wedge \langle u \rangle \widehat{w'}$ is consistent. Applying $(A2)$, $\widehat{w} \wedge \Diamond\widehat{w'}$ is consistent.

Alternately, if $w \xrightarrow{d} w'$, then again $\widehat{w} \wedge \Diamond\widehat{w'}$ is consistent.

By part (i) of Proposition 2.3, $\vdash \widehat{w'} \supset \Diamond\alpha$. By $(DR2)$, we get $\vdash \Diamond\widehat{w'} \supset \Diamond\Diamond\alpha$, and by axiom $(A2)$, it follows that $\vdash \Diamond\widehat{w'} \supset \Diamond\alpha$. Hence $\widehat{w} \wedge \Diamond\alpha$ is consistent. Since $\Diamond\alpha \in w'$, it is in CL and using Proposition 2.3(ii), we get $\Diamond\alpha \in w$. □

Define now the model $M_0 = (TS_0, V_0)$ where for every $w \in AT$, $V_0(w) = w \cap P$. Since α_0 is consistent, it must belong to some atom w_0 in AT, hence by the following lemma it is satisfiable in M_0.

Lemma 2.6 $\forall\beta \in CL : \forall w \in AT : M_0, w \models \beta$ *iff* $\beta \in w$.

Proof: We proceed by induction on the structure of β. If $\beta \in P$ or β is of the form $\sim\alpha$ or $\alpha_1 \vee \alpha_2$ the proof is routine. Hence assume that β is of the form $\langle u \rangle \alpha$.

Suppose $\langle u \rangle \alpha \in w$. Then $\widehat{w} \wedge \langle u \rangle \alpha$ is consistent by part (ii) of Proposition 2.3. This implies that $\langle u \rangle \alpha$ is consistent and by derived rule (uG), α is consistent. Then the set $W = \{w' \mid \alpha \in w'\}$ of atoms is nonempty and by PC, $\vdash \alpha \supset \widetilde{W}$. By $(DR1)$, we can then deduce that $\vdash \langle u \rangle \alpha \supset \langle u \rangle \widetilde{W}$. Thus $\widehat{w} \wedge \langle u \rangle \widetilde{W}$ is consistent. By thesis $(T6)$, there exists $w' \in W$ such that $\widehat{w} \wedge \langle u \rangle \widehat{w'}$ is consistent. By definition of $\rightarrow$, we then obtain $w \xrightarrow{u} w'$. Since $\alpha \in w'$, it must be the case that $M, w' \models \alpha$ by the induction hypothesis. Hence $M, w \models \langle u \rangle \alpha$.

Next suppose that $M, w \models \langle u \rangle \alpha$. Then there exists $w' \in AT$ such that $w \xrightarrow{u} w'$ and $M, w' \models \alpha$. By the induction hypothesis, $\alpha \in w'$. By definition of $\rightarrow$, $\widehat{w} \wedge \langle u \rangle \widehat{w'}$ is consistent. Thesis $(T5)$ implies $\widehat{w} \wedge \langle u \rangle \alpha$ is consistent. Since $\langle u \rangle \alpha \in CL$, we get $\langle u \rangle \alpha \in w$ from part (ii) of Proposition 2.3.

Now consider the case where β is of the form $\Diamond\alpha$. Suppose $\Diamond\alpha \in w$. Then $\widehat{w} \wedge \Diamond\alpha$ is consistent. Hence $\Diamond\alpha$ and α are consistent. Define $W = \{w' \mid \alpha \in w'\}$ as above. Again, using the derived rule $(DR2)$ and thesis $(T7)$, we will get $w' \in W$ such that $\widehat{w} \wedge \Diamond\widehat{w'}$ is consistent. Consequently $w \xrightarrow{d} w'$. Using the induction hypothesis, $M, w' \models \alpha$ and therefore $M, w \models \Diamond\alpha$.

Finally suppose that $M, w \models \Diamond\alpha$. Then there is a $w' \in \mathcal{R}(w)$ such that $M, w' \models \alpha$. By the induction hypothesis, $\alpha \in w'$, and by thesis $(T10)$, $\Diamond\alpha \in w'$. By repeated application of Lemma 2.5, we now get $\Diamond\alpha \in w$, as required. □

Theorem 2.7 (Completeness) *If* $\models \alpha$ *then* $\vdash \alpha$.

Proof: As observed earlier, Lemma 2.6 at once implies that every consistent formula is satisfiable. □

Observe that we can in fact obtain a model based on a *pointed* frame by taking $TS_1 = (\mathcal{R}_{TS_0}(w_0), \rightarrow, w_0)$ in the above construction. Hence the completeness result holds for models based on pointed transition systems as well. We can in fact extract a more important result. By soundness of the axiom system, if α is satisfiable then it is consistent. The proof of the completeness theorem then guarantees that it has a model of size at most 2^{cn} where $c > 0$ is a constant and n is the length of α. Hence we have:

Theorem 2.8 (Decidability) *Satisfiability in our logic is decidable in nondeterministic exponential time.*

The standard filtration technique [FL] can also be applied to get a direct, model-theoretic proof of decidability [Parikh].

A remark about the upper and lower bounds. It is easy to see that the Fischer-Ladner lower bound of *deterministic* exponential time for PDL [FL] will also hold for our logic. The same proof goes through except for the fact that the $[\vdash^*]$ of page 207, line 12 [FL] must be replaced by $\Box$. As for the upper bound, PDL has been shown by Pratt [Pra80] to be decidable in deterministic exponential time. However, unlike PDL, our models are built from (hyper-)cubes corresponding to the relation $\stackrel{u}{\rightarrow}$. Such cubes can be exponential in the size of the formula and it is not obvious that guessing them can be avoided.

3 Event Structures and Net Systems

The soundness and completeness theorems of the previous section can be together viewed as a logical characterization of the class of distributed transition systems. We mean this in the spirit of a result such as "S4 is sound and complete w.r.t the class of partial orders" in modal logic [HC]. Here we wish to show that our axiomatization also characterizes finitary event structures and elementary net systems.

As shown in Section 1, there is a natural way of associating a dts TS_{ES} with every Σ-labelled event structure. Similarly there is a dts $TS_{\mathcal{N}}$ associated with every Σ-labelled elementary net system. It is easy to see that there are dts's which cannot be generated by event structures or net systems.

For instance, let a, b, c be three distinct letters in Σ. Figure 7 shows a dts which can not be isomorphic (in the obvious sense) to a dts associated with a Σ-labelled event structure. The events corresponding to the $\{a, c\}$-cube are dependent on the a and c events performed in state s_λ, which are in conflict. In an event structure, concurrent events cannot be dependent on conflicting ones.

A similar claim can be made for dts's associated with elementary net systems.

On the other hand, let SAT, SAT_{ES} and SAT_{NS} be the set of formulas satisfiable by models based on all dts's, those based on dts's associated with labelled event structures and those based on dts's associated with labelled net systems respectively. We show in this section that $SAT_{ES} = SAT_{NS} = SAT$. That is, (satisfiable) formulas of our logic can be satisfied by dts's associated with event structures and net systems.

To prove this result, we make use of the standard notion of bisimulation [Park].

Definition 3.1 *Let $TS_i = (S_i, \rightarrow_i)$, $i = 1, 2$, be two dts's over Σ. Then a* **step bisimulation** *between them is a relation $R \subseteq S_1 \times S_2$ such that $s_1 \ R \ s_2$ implies:*

- *If $s_1 \overset{u}{\rightarrow}_1 s_1'$ then there exists $s_2' \in S_2$ such that $s_2 \overset{u}{\rightarrow}_2 s_2'$ and $s_1' \ R \ s_2'$.*
- *If $s_2 \overset{u}{\rightarrow}_2 s_2'$ then there exists $s_1' \in S_1$ such that $s_1 \overset{u}{\rightarrow}_1 s_1'$ and $s_1' \ R \ s_2'$.*

Proposition 3.2 *Let $M_i = ((S_i, \rightarrow_i), V_i)$, $i = 1, 2$ be two models and R a step bisimulation between $(S_1, \rightarrow_1)$ and $(S_2, \rightarrow_2)$ such that $s_1 \ R \ s_2$ implies $V_1(s_1) = V_2(s_2)$ for every s_1, s_2. Then for every α and every $(s_1, s_2) \in R$,*

$$M_1, s_1 \models \alpha \text{ iff } M_2, s_2 \models \alpha.$$

Proof: We can first show that $s_1 \ R \ s_2$ implies:

- If $s_1' \in \mathcal{R}_{TS_1}(s_1)$ then there exists $s_2' \in \mathcal{R}_{TS_2}(s_2)$ such that $s_1' \ R \ s_2'$.
- If $s_2' \in \mathcal{R}_{TS_2}(s_2)$ then there exists $s_1' \in \mathcal{R}_{TS_1}(s_1)$ such that $s_1' \ R \ s_2'$.

Then structural induction on α will give the result. For instance, one can follow the proof of the p-morphism theorem in [HC]. $\Box$

Lemma 3.3 *Let $TS = (S, \rightarrow, s_0)$ be a countable pointed dts over Σ. Then there exists a labelled event structure ES and a step bisimulation R between $TS_{ES} = (C_{ES}, \Rightarrow_{ES})$ and TS such that $\emptyset \ R \ s_0$.*

Proof: Fix a countably infinite set of events $\widehat{E}$ and fix an enumeration of $\wp_{fin}(\widehat{E}) \times S \times \wp_{fin}(\Sigma) \times S$. Since $\widehat{E}$, Σ and S are countable such an enumeration exists. We will inductively construct an infinite sequence (ES_0, R_0), (ES_1, R_1), ... such that for every $i \geq 0$,

1. $ES_i = (E_i, \leq_i, \#_i, \phi_i)$ is a finite labelled event structure whose events are members of $\widehat{E}$.
2. $E_i \subseteq E_{i+1}$, $\leq_{i+1} \lceil E_i = \leq_i$, $\#_{i+1} \lceil E_i = \#_i$ and $\phi_{i+1} \lceil E_i = \phi_i$.
3. If $e_1 \ co_{ES_i} \ e_2$ then $\forall j \leq i : e_1 \in E_j$ iff $e_2 \in E_j$.
4. $R_i \subseteq C_{ES_i} \times S$ with the property $\emptyset \ R_i \ s_0$, R_i is a function and $R_i = R_{i+1} \lceil (C_{ES_i} \times S)$.

We will abbreviate C_{ES_i} by C_i, $\Rightarrow_{ES_i}$ by $\Rightarrow_i$, etc. The tuple (c, s, u, s') is a **requirement** for (ES_i, R_i) if $c \in C_i$, $c \ R_i \ s$ and $s \overset{u}{\rightarrow} s'$ in TS. The requirement is **live** if there is no $c' \in C_i$ such that $c \overset{u}{\Rightarrow}_i c'$ and $c' \ R_i \ s'$.

The "limit" of this sequence will be (ES, R), the event structure and bisimulation required by the lemma.

Set $ES_0 = (\emptyset, \emptyset, \emptyset, \emptyset)$, so that $TS_0 = (\{\emptyset\}, \Rightarrow_0)$ where $\Rightarrow_0 = \{(\emptyset, \emptyset, \emptyset)\}$. Set $R_0 = \{(\emptyset, s_0)\}$. Clearly (ES_0, R_0) satisfies the inductive conditions.

Assume that (ES_i, R_i) have been defined for $i \geq 0$ satisfying the required properties.

If there are no live requirements at stage i, set $(ES_{i+1}, R_{i+1}) = (ES_i, R_i)$. Otherwise pick the live requirement (c, s, u, s') with the least index in the enumeration of $\wp_{fin}(\widehat{E}) \times S \times \wp_{fin}(\Sigma) \times S$ we have fixed. Note that $u \neq \emptyset$ because $s \xrightarrow{\emptyset} s'$ implies $c \stackrel{\emptyset}{\Rightarrow}_i c$ and $c\ R_i\ s'$. Let $u = \{a_1, a_2, \ldots, a_n\}$. Pick $Y = \{e_1, e_2, \ldots, e_n\}$ from $\widehat{E} - E_i$. Since $\widehat{E}$ is countable and E_i is finite, we can always find such a Y.

Define $ES_{i+1} = (E_{i+1}, \leq_{i+1}, \#_{i+1}, \phi_{i+1})$ by

$$
\begin{array}{lcl}
E_{i+1} & \stackrel{\text{def}}{=} & E_i \cup Y \\
\leq_{i+1} & \stackrel{\text{def}}{=} & \leq_i \cup (c \times Y) \cup \{(e,e) \mid e \in Y\} \\
\#_{i+1} & \stackrel{\text{def}}{=} & \#_i \cup ((E_i - c) \times Y) \cup (Y \times (E_i - c)) \\
\phi_{i+1}(e) & \stackrel{\text{def}}{=} & \begin{cases} a_j, & \text{for } e = e_j,\ 1 \leq j \leq n \\ \phi_i(e), & \text{for } e \in E_i \end{cases}
\end{array}
$$

First observe that the inductive conditions (1), (2) and (3) hold. We have

$$co_{i+1} = co_i \cup \{(e, e') \mid e, e' \in Y,\ e \neq e'\}.$$

ES_{i+1} is finite and Σ-labelled (preserving concurrency), but we have to verify that it is an event structure.

$\leq_{i+1}$ is clearly reflexive. It is transitive because c is downward closed. Since $\leq_i$ was antisymmetric, $\leq_{i+1}$ is antisymmetric by definition.

To prove conflict inheritance, let $e\ \#_{i+1}\ e' \leq_{i+1} e''$. Suppose $e'' \in Y$. Then, by definition of $\leq_{i+1}$, either $e' = e''$, in which case we are done, or $e' \in c$. But then, e must be in E_i and not in c. Hence $e\ \#_{i+1}\ e''$ by definition.

Suppose that $e'' \in E_i$. Now e' must be in E_i. If e is also in E_i, since ES_i is an event structure, we have $e\ \#_i\ e''$ and hence $e\ \#_{i+1}\ e''$. Otherwise, e is in Y. Hence $e' \in E_i - c$, whence $e'' \notin c$ as well. By definition, $(e, e'') \in \#_{i+1}$.

This completes the construction of ES_{i+1}. Finally observe that

$$C_{i+1} = C_i \cup \{c \cup y \mid y \subseteq Y\}$$

Since $s \xrightarrow{u} s'$, let f be a u-cube defined by it. Let

$$R_{i+1} = R_i \cup \{(c \cup y, f(\phi_{i+1}(y))) \mid y \subseteq Y\}$$

By taking the componentwise union of the (ES_i, R_i)'s, we obtain the required pair (ES, R). ES is finitary since each event in it comes from some ES_j, $j \geq 0$ which is finite, and since $\leq_i \lceil E_j = \leq_j$, for $i \geq j$, the "past" of each event is finite. We use inductive condition (3) to establish that R is a step bisimulation.

In one direction, suppose $c\ R\ s$ and $c \stackrel{u}{\Rightarrow}_{ES} c'$. Then there is a set of concurrent events y such that $c' = c \cup y$ and $\phi(y) = u$. By (3), there is a minimum $j \geq 0$ such that $y \subseteq E_{j+1} - E_j$. Let

(c, s, v, s') be the requirement chosen at ES_j to obtain ES_{j+1} and $f : \wp(v) \to S$ be the chosen v-cube. By definition of C_{i+1}, it must be the case that $\exists v_0 : v_0 \cup u \subseteq v : c' = c \cup v_0,\ c'' = c \cup v_0 \cup u$. Then $c' \ R \ f(v_0)$, $c'' \ R \ f(v_0 \cup u)$ and $f(v_0) \xrightarrow{u} f(v_0 \cup u)$ in TS.

For the other direction, suppose $s_1 \xrightarrow{u} s_2$ and $c \ R \ s_1$. Since (c, s_1, u, s_2) is a fulfilled requirement, there exists a c' such that $c' \ R \ s_2$ and $c \overset{u}{\Rightarrow}_{ES} c'$. □

Theorem 3.4 $SAT = SAT_{ES}$.

Proof: One direction is trivial. For the other, suppose $\alpha \in SAT$. Then there is a countable (in fact, finite) pointed model $M = (TS, V)$, $TS = (S, \to, s_0)$ such that $M, s_0 \models \alpha$. By the previous lemma, there is an event structure ES with a step bisimulation $R \subseteq C_{ES} \times S$ such that $\emptyset \ R \ s_0$. The proof of the lemma also establishes that R is a function, hence we can define $V_{ES}(c) = V(s)$, where $c \ R \ s$. Let $M_{ES} = (TS_{ES}, V_{ES})$. By Proposition 3.2, $M_{ES}, \emptyset \models \alpha$. Hence $\alpha \in SAT_{ES}$. □

Thus we have that satisfiability over the class of dts's generated by event structures is also decidable in nondeterministic exponential time and the axiom system of Section 2 is sound and complete for this class.

We can similarly characterize the state space of elementary net systems. The crucial step is again provided by establishing a step bisimulation. In this case, we can use the work of [NPW] and provide a bijection.

Lemma 3.5 *Let $ES = (E, \leq, \#, \phi)$ be a labelled event structure. Then there exists a labelled net system $\mathcal{N} = (B, E, F, c_{in}, \phi)$ and a bijection $h : TS_{ES} \to TS_{\mathcal{N}}$ such that for every configuration $x \in C_{ES}$,*

$$x \overset{u}{\Rightarrow}_{ES} x' \text{ iff } h(x) \overset{u}{\Rightarrow}_{\mathcal{N}} h(x').$$

Proof: Set $B = B_{id} \cup B_< \cup B_\#$ where

- $B_{id} = \{\{e\}\} \mid e \in E\}$
- $B_< = \{(e_1, e_2) \mid e_1 \leq e_2,\ e_1 \neq e_2\}$
- $B_\# = \{\{e, e'\} \mid e \# e'\}$.

Next set $F = F_{id} \cup F_< \cup F_\#$ where

- $F_{id} = \{(\{e\}, e) \mid \{e\} \in B_{id}\}$
- $F_< = \{(e_1, (e_1, e_2)), ((e_1, e_2), e_2) \mid (e_1, e_2) \in B_<\}$
- $F_\# = \{(\{e, e'\}, e), (\{e, e'\}, e') \mid \{e, e'\} \in B_\#\}$

Finally set $c_{in} = B_{id} \cup B_{\#}$.

Then $\mathcal{N} = (B, E, F, c_{in}, \phi)$ is a labelled net system. To see this, we need to verify that

Claim : $e_1 \ Ind_N \ e_2$ iff $e_1 \ co_{ES} \ e_2$.

We leave the verification of this claim to the reader, as well as that of the fact that $h : C_{ES} \to \mathcal{C}_{\mathcal{N}}$ defined by

$$h(x) \stackrel{\text{def}}{=} (c_{in} \cup x^{\bullet}) - {}^{\bullet}x.$$

is the required bijection. □

Theorem 3.6 $SAT = SAT_{NS}$.

Proof: One direction is trivial. For the other, by Theorem 3.4, $SAT \subseteq SAT_{ES}$. The bijection h defines a step bisimulation. By defining a valuation function and using Proposition 3.2, $SAT_{ES} \subseteq SAT_{NS}$. □

4 Deterministic distributed transition systems

In this section we begin the study of deterministic dts's from the vantage point of our logic. We begin by introducing terminology that will be used throughout the rest of the paper. As before, by a dts we will mean a dts over Σ where Σ is a countably infinite set of actions.

Definition 4.1 *Let $TS = (S, \Sigma, \rightarrow)$ be a dts.*

1. *TS is said to be* **deterministic** *if*

$$\forall s, s_1, s_2 \in S, \forall u \in \wp_{fin}(\Sigma) : s \xrightarrow{u} s_1 \text{ and } s \xrightarrow{u} s_2 \text{ implies } s_1 = s_2.$$

2. *The model $M = (TS, V)$ is said to be deterministic if TS is.*

3. *α is said to be* **deterministically satisfiable** *if there exists a deterministic model $M = ((S, \rightarrow), V)$ and $s \in S$ such that $M, s \models \alpha$.*

4. *α is said to be* **deterministically valid** *if $M, s \models \alpha$ for every deterministic model $M = ((S, \rightarrow), V)$ and every $s \in S$. We write $\models_{Det} \alpha$ to denote that α is deterministically valid.*

5. *DSAT and DVAL denote the set of deterministically satisfiable and deterministically valid formulas respectively.*

Deterministic dts's arise naturally in a number of ways. Let $ES = (E, \leq, \#)$ be a finitary event structure. Then $(C_{ES}, \rightarrow_{ES})$ is a deterministic dts over E. Similarly, if $\mathcal{N} = (B, E, F, c_{in})$ is an elementary net system then $(\mathcal{C}_{\mathcal{N}}, \rightarrow_{\mathcal{N}})$ is a deterministic dts over E. For Σ-labelled event structures and net systems one can place well motivated restrictions on the labelling functions to ensure that the associated dts's over Σ are deterministic. We will not go into details here.

It turns out that, from a logical point of view, determinacy adds a great deal of expressive power. One of our aims is to bring this out in a number of ways in this and subsequent sections.

First we consider the simple-looking formula $\Box\langle\{x,y\}\rangle True$. Any deterministic model of this formula must contain the grid $\mathbf{N} \times \mathbf{N}$ shown in Figure 8. (A formal proof is provided in the next section.) We have omitted the set arrows $\xrightarrow{\{x,y\}}$ in the picture.

Secondly, we can point out that for the class of deterministic models, there is no hope of getting "equivalent" deterministic models based on event structures or net systems (in the sense of the translation theorems of the previous section). To see this, observe that the formula $\langle\{a,b\}\rangle True \wedge \langle\{b,c\}\rangle True \wedge [\{a,c\}]False \wedge [b]\langle\{a,c\}\rangle True$ is in $DSAT$ because we can find a deterministic model for it, based on the dts shown in Figure 7. However, we know that the dts cannot be generated by an event structure. The bisimulation technique of Section 3 is not of use because we are restricted to deterministic systems. Hence the class of deterministic dts models strictly includes those based on finitary prime event structures. Once again, a similar claim can be made for deterministic models based on elementary net systems.

Another piece of evidence supporting the view that determinacy is very expressive is provided by the axiomatization of $DVAL$ which we present now.

Axiom System D

Axiom schemes

$(A0)$	All the substitutional instances of the tautologies of PC
$(A1)$	$\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$
$(A2)$	$\Box\alpha \supset [u]\alpha \wedge \Box\Box\alpha$
$(A3)$	$[u](\alpha \supset \beta) \supset ([u]\alpha \supset [u]\beta)$
$(A4)$	$\alpha \equiv \langle\emptyset\rangle\alpha$
$(A5)$	$\langle u\rangle\alpha \supset \langle v\rangle\langle u-v\rangle\alpha$, for $v \subseteq u$
$(A6)$	$\langle u\rangle\alpha \supset [u]\alpha$

Inference rules

$(MP)\ \dfrac{\alpha, \alpha \supset \beta}{\beta} \quad (TG)\ \dfrac{\alpha}{\Box\alpha}$

The characteristic axiom of this system is the determinacy axiom (A6). In its presence the rule (*Step*) of Section 2 can be replaced by the much simpler (A5) of the present system. It can be shown that (*Step*) is a derived inference rule in the axiom system.

We let $\vdash_D \alpha$ denote the fact that α is a thesis of the system D. In this section we will, for convenience, write $\vdash \alpha$ to mean $\vdash_D \alpha$ and say α is *consistent* to mean that it is consistent w.r.t. the system D. From the definitions, we easily have:

Theorem 4.2 (Soundness) *If* $\vdash_D \alpha$ *then* $\models_{Det} \alpha$.

The completeness argument will be a lot more involved than the one presented in Section 2. A simple reason is that the "filtration" technique used in the earlier proof will produce, in general, nondeterministic models. A deeper reason is that, as we will prove in the next section, DVAL is not a recursive set.

We will use the following theses and derived rules:

Theses

$(T1)$ $[u]\alpha \wedge \langle u\rangle\beta \supset \langle u\rangle(\alpha \wedge \beta)$
$(T2)$ $\Box\alpha \wedge \Diamond\beta \supset \Diamond(\alpha \wedge \beta)$
$(T3)$ $[u](\alpha \wedge \beta) \equiv [u]\alpha \wedge [u]\beta$
$(T4)$ $\Box(\alpha \wedge \beta) \equiv \Box\alpha \wedge \Box\beta$
$(T5)$ $\Box\alpha \supset \alpha$
$(T6)$ $\Box\alpha \supset [u]\Box\alpha$
$(T7)$ $\langle u\rangle True \supset \langle v\rangle True,\quad$ for $v \subseteq u$
$(T8)$ $\langle u\rangle\alpha \supset [v]\langle u-v\rangle\alpha,\quad$ for $v \subseteq u$
$(T9)$ $\langle u\rangle True \wedge [u]\alpha \supset [v][u-v]\alpha,\quad$ for $v \subseteq u$

Derived rules

$$(uG)\ \frac{\alpha}{[u]\alpha} \qquad (DR1)\ \frac{\alpha \supset \beta}{\langle u\rangle\alpha \supset \langle u\rangle\beta} \qquad (DR2)\ \frac{\alpha \supset \beta}{[u]\alpha \supset [u]\beta}$$

The derivations are straightforward and we once again omit the details.

A number of new notions will be needed for the completeness proof. The dts $(S, \Sigma, \rightarrow)$ is said to be:

- *sequential* iff $\forall s, s' \in S : \forall u \in \wp_{fin}(\Sigma) : s \xrightarrow{u} s'$ implies $|u| \leq 1$.
- *finite* iff both S and $\rightarrow$ are finite sets.
- *acyclic* iff $\forall s, s' \in S : s \in \mathcal{R}(s')$ and $s' \in \mathcal{R}(s)$ implies $s = s'$.

If a pointed dts $TS = (S, \rightarrow, s_0)$ is acyclic then s_0 is said to be the *root* of TS.

Now assume that TS is acyclic and has root s_0. Then we can define the function $depth_{TS} : S \rightarrow \mathbf{N}$ as follows:

$$depth_{TS}(s_0) = 0$$

$$depth_{TS}(s) = max\{depth_{TS}(s') \mid (s', a, s) \in \rightarrow\} + 1$$

We will omit the subscript and refer to the *depth* function when the dts on which it is defined is understood. We say that a rooted acyclic dts is *graded* if $depth(s') = 1 + depth(s)$ for every (s, a, s') in the transition relation. Informally, every path from the root to a state s must be of *equal* length in a graded dts.

Next we need the notion of a thin u-cube. Let $TS = (S, \Sigma, \rightarrow)$ be a dts and $s, s' \in S$. Then a *thin u-cube* (from s to s') is a function $f \in \mathcal{F}[u, S]$ which satisfies:

- $f(\emptyset) = s$ and $f(u) = s'$
- $\forall v \subseteq u : \forall a \in u : f(v - \{a\}) \xrightarrow{a} f(v \cup \{a\})$.

Note that the existence of a u-cube implies that of a thin u-cube. The converse, in general, is not true.

By an MCS, we mean a maximal consistent subset of the set of all formulas of Step-TL (that is, a consistent set which is not properly included in any consistent set). Of course, consistency is now relative to the axiom system D.

Definition 4.3 *A* **chronicle structure** *is a pair $CH = (TS, T)$ where $TS = (S, \rightarrow)$ is a deterministic sequential dts and T is a map (called the* **chronicle***) which assigns an MCS to each s in S.*

1. *Let α_0 be a formula. T is said to be α_0-***coherent** *in CH iff $\forall s, s' \in S$:*
   - *(a) If $a \in Voc(\alpha_0), s \xrightarrow{a} s'$ and $[a]\alpha \in T(s)$ then $\alpha \in T(s')$.*
   - *(b) If $s' \in \mathcal{R}(s)$ and $\Box\alpha \in T(s)$ then $\alpha \in T(s')$.*
2. *A* **live successor requirement** *for α_0 in CH is a pair $(s, \langle a \rangle True)$ where $s \in S, a \in Voc(\alpha_0), \langle a \rangle True \in T(s)$ and there is no $s' \in S$ such that $s \xrightarrow{a} s'$.*
3. *A* **live future requirement** *for α_0 in CH is a pair $(s, \Diamond\alpha)$ where $s \in S, \Diamond\alpha \in T(s) \cap CL(\alpha_0)$ and for all $s' \in \mathcal{R}(s), \alpha \notin T(s')$.*
4. *Let α_0 be a formula. CH is said to be α_0-***perfect** *iff T is α_0-coherent in CH and the following conditions hold:*
   - *(a) There exists $s_0 \in S$ such that $\alpha_0 \in T(s_0)$.*
   - *(b) There are no live successor requirements for α_0 in CH.*
   - *(c) There are no live future requirements for α_0 in CH.*

As before, we omit the parameter α_0 when clear from the context. The following observation about α_0-coherent chronicles will prove useful later.

Proposition 4.4 *Let T be α_0-coherent in the chronicle structure $CH = (TS, T)$ where $TS = (S, \rightarrow)$. Let $s \in S$ and $u \subseteq Voc$ such that $\langle u \rangle True \in T(s)$ and there exists a thin u-cube from s to s' for some s' in S. Then $\{\alpha \mid [u]\alpha \in T(s)\} \subseteq T(s')$.*

Proof: The proof proceeds by induction on size of u.

base: $|u| = 0$. Then $u = \emptyset$ and hence $s = s'$ and the result follows by Axiom A4.

step: $|u| = k > 0$. Let $a \in u$ and $[u]\alpha \in T(s)$. By definition of thin cubes, $f(u - \{a\}) \xrightarrow{a} f(u)$ and there exists a thin $(u - \{a\})$-cube between s and $f(u - \{a\})$. It suffices to prove that $[a]\alpha \in T(f(u - \{a\}))$ as the result would then follow by α_0-coherence of T. Now $\langle u \rangle True \in T(s)$ and hence by thesis $(T7)$, $\langle u - \{a\} \rangle True \in T(s)$ as well. Then thesis $(T9)$ gives $[u - \{a\}][a]\alpha \in T(s)$. By induction hypothesis, we get $[a]\alpha \in T(f(u - \{a\}))$, as required. □

Now we show that a model can be "pulled out" from an α_0-perfect chronicle structure. This technique is due to Burgess [Bur].

Lemma 4.5 *Suppose α_0 is a consistent formula and $CH = (TS, T)$ is α_0-perfect. Then $\alpha_0 \in DSAT$.*

Proof: Let $TS = (S, \rightarrow)$. Define $TS' = (S, \Rightarrow)$ by

$$\Rightarrow \stackrel{\text{def}}{=} \rightarrow \cup \{(s, u, s') \mid u \subseteq Voc, \langle u \rangle True \in T(s)$$

$$\text{and there is a thin } u\text{-cube from } s \text{ to } s' \text{ in } TS\}$$

Claim : TS' is a dts.

Whenever $s \stackrel{u}{\Rightarrow} s'$, we need to show that there is a u-cube between s and s'. If $s \stackrel{u}{\rightarrow} s'$ then $|u| \leq 1$ and there exists a thin u-cube from s to s' in TS. Since $|u| \leq 1$ and $\rightarrow \subseteq \Rightarrow$, there is a u-cube from s to s' in TS' as well.

Otherwise $\langle u \rangle True \in T(s), u \subseteq Voc$ and there is a thin u-cube f between s and s' in TS. Then for every $v_1 \subseteq v_2 \subseteq u$, there is a thin v_1-cube between s and $f(v_1)$ and a thin $(v_2 - v_1)$-cube between $f(v_1)$ and $f(v_2)$. Since $\langle u \rangle True \in T(s)$, by thesis $(T8)$ $[v_1]\langle v_2 - v_1 \rangle True \in T(s)$ and hence by Proposition 4.4, $\langle v_2 - v_1 \rangle True \in T(f(v_1))$. Thus, by definition of $\Rightarrow$, $f(v_1) \stackrel{v_2 - v_1}{\Longrightarrow} f(v_2)$. Clearly f is a u-cube between s and s'.

Claim : TS' is a deterministic dts.

Suppose $s \stackrel{u}{\Rightarrow} s_1$ and $s \stackrel{u}{\Rightarrow} s_2$. If $|u| \leq 1$, the result follows by determinacy of TS. Otherwise let f be a thin u-cube between s and s_1 and g a thin u-cube between s and s_2. We show by induction on v that $f(v) = g(v)$. The base case is trivial. For the induction step, let $s' = f(v - \{a\}) = g(v - \{a\}), a \in u$. We have $s' \stackrel{a}{\rightarrow} f(v)$ and $s' \stackrel{a}{\rightarrow} g(v)$. By determinacy of TS, we have $f(v) = g(v)$ as required.

Thus TS' is a frame. Define the model $M = (TS', V)$ by $V(s) \stackrel{\text{def}}{=} T(s) \cap P \cap CL$. Then M is a model based on a deterministic dts.

Since CH is α_0-perfect, there exists $s_0 \in S$ such that $\alpha_0 \in T(s_0)$. The following claim proves that $M, s_0 \models \alpha_0$ and hence that $\alpha_0 \in DSAT$.

Claim : $\forall \gamma \in CL : \forall s \in S : \gamma \in T(s)$ iff $M, s \models \gamma$.

The proof is by induction on the structure of γ.

Base: When $\gamma \in P$, the result follows by definition of V.

Step: The cases where γ is of the form $\sim\delta$ or $\delta_1 \vee \delta_2$ are routine.

Case $\gamma \equiv \langle u \rangle \alpha$: Suppose $\langle u \rangle \alpha \in T(s)$. As $\vdash \alpha \supset True$, by rule $(DR2)$, $\langle u \rangle True \in T(s)$. Further by axiom $(A6)$, $[u]\alpha \in T(s)$. Since CH is α_0-perfect, there exists $s' \in S$ such that there is a thin u-cube from s to s'. By Proposition 4.4, $\alpha \in T(s')$. By the induction hypothesis, $M, s' \models \alpha$. Further, by definition of $\Rightarrow$, $s \stackrel{u}{\Rightarrow} s'$. Thus $M, s \models \langle u \rangle \alpha$.

Suppose $M, s \models \langle u \rangle \alpha$. Let $s' \in S$ such that $s \stackrel{u}{\Rightarrow} s'$ and $M, s' \models \alpha$. By the induction hypothesis, $\alpha \in T(s')$. If $\langle u \rangle \alpha \in T(s)$, we are done. Otherwise, $[u]{\sim}\alpha \in T(s)$. If $|u| \leq 1$, we get ${\sim}\alpha \in T(s')$ by α_0-coherence of T, since $u \subseteq Voc$. Otherwise $\langle u \rangle True \in T(s)$ and there is a thin u-cube between s and s'. Again by Proposition 4.4, we get ${\sim}\alpha \in T(s')$, contradicting consistency of $T(s')$.

Case $\gamma \equiv \Diamond\alpha$: Suppose $\Diamond\alpha \in T(s)$. As CH is α_0-perfect, there exists $s' \in \mathcal{R}_{TS'}(s)$ such that $\alpha \in T(s')$. By the induction hypothesis, $M, s' \models \alpha$ and hence $M, s \models \Diamond\alpha$.

Suppose $M, s \models \Diamond\alpha$. Let $s' \in \mathcal{R}_{TS'}(s)$ such that $M, s' \models \alpha$. By the induction hypothesis, $\alpha \in T(s')$. If $\Diamond\alpha \in T(s)$, we are done. Otherwise, $\Box{\sim}\alpha \in T(s)$. Since $\mathcal{R}_{TS}(s) = \mathcal{R}_{TS'}(s)$, by α_0-coherence of T, we get ${\sim}\alpha \in T(s')$, contradicting the consistency of $T(s')$. □

Thus, given a consistent formula α_0, we need to construct an α_0-perfect chronicle structure. The following results will prove to be useful in the construction.

Proposition 4.6 *Let A be an MCS such that $\Diamond\alpha \in A$. Then there exists an MCS B such that $\{\beta \mid \Box\beta \in A\} \subseteq B$ and $\alpha \in B$.*

Proof: Let $\Gamma = \{\beta \mid \Box\beta \in A\} \cup \{\alpha\}$.

Consider a finite subset of Γ, say $\Gamma' = \{\beta_1, \ldots, \beta_k, \alpha\}$. Then $\{\Box\beta_1, \ldots, \Box\beta_k, \Diamond\alpha\} \subseteq A$ and A is an MCS, hence $\Box\beta_1 \wedge \ldots \wedge \Box\beta_k \wedge \Diamond\alpha \in A$. By thesis $(T3)$, we get $\Box(\beta_1 \wedge \ldots \wedge \beta_k) \wedge \Diamond\alpha \in A$. By thesis $(T4)$, the formula $\Diamond(\beta_1 \wedge \ldots \wedge \beta_k \wedge \alpha)$ is in A and is consistent. By rule (TG), $\beta_1 \wedge \ldots \wedge \beta_k \wedge \alpha$ is consistent, that is, Γ' is consistent. Since any arbitrary finite subset of Γ is consistent, so is Γ. Let B be any MCS such that $\Gamma \subseteq B$. B is the required MCS. □

Proposition 4.7 *Let A be an MCS such that $\langle a\rangle\alpha \in A$, for some $a \in \Sigma$. Then there exists an MCS B such that $\{\beta \mid [a]\beta \in A\} \subseteq B$ and $\alpha \in B$.*

Proof: Similar to that of the above Proposition. □

Lemma 4.8 *Let α_0 be a consistent formula. Then there exists an α_0-perfect chronicle structure.*

Proof: Fix $\widehat{S} = \{\widehat{s_0}, \widehat{s_1}, \ldots\}$ a countable set.

We define a sequence of chronicle structures $CH_k = (TS_k, T_k), k \geq 0$, where $TS_k = (S_k, \rightarrow_k)$, such that the following conditions hold:

(A) TS_k is a finite, pointed, acyclic, graded deterministic dts with root $\widehat{s_0}$,

(B) T_k is an α_0-coherent chronicle in CH_k,

(C) $\rightarrow_k = \rightarrow_{k+1} \lceil S_k$ and $T_k = T_{k+1} \lceil S_k$.

We will use $depth_k$ to denote the function $depth_{TS_k}$. Further for all k, we define $\doteq_k \subseteq (S_k \times Voc) \times (S_k \times Voc)$ as follows:

$(s, a) \doteq_k (s', b)$ iff

- $a \neq b$,
- $depth_k(s) = depth_k(s')$,
- there exists $s'' \in S_k$ such that $\langle\{a,b\}\rangle True \in T_k(s'')$, $s'' \xrightarrow{b}_k s$ and $s'' \xrightarrow{a}_k s'$, and
- $(s, \langle a\rangle True)$ and $(s', \langle b\rangle True)$ are live successor requirements in CH_k.

Define $=_k$ to be $(\doteq_k)^*$. $\doteq_k$ is irreflexive and symmetric. $=_k$ is the equivalence relation we will use. The idea is that when we satisfy any successor requirement, in order to ensure determinacy, we satisfy all equivalent successor requirements.

The construction proceeds by induction on k. For the base case, set $TS_0 = (S_0, \rightarrow_0)$, where $S_0 \stackrel{\text{def}}{=} \{\widehat{s_0}\}$ and $\rightarrow_0 \stackrel{\text{def}}{=} \{(\widehat{s_0}, \emptyset, \widehat{s_0})\}$. Since α_0 is consistent there exists an MCS A such that $\alpha_0 \in A$. Set $T_0(s_0) \stackrel{\text{def}}{=} A$. It can be easily checked that $CH_0 = (TS_0, T_0)$ is a chronicle structure satisfying the conditions (A), (B) and (C).

Inductively let $CH_k = (TS_k, T_k)$ be given satisfying the inductive conditions. If CH_k has no live requirements, set $CH_{k+1} \stackrel{\text{def}}{=} CH_k$. Otherwise pick a $depth_k$-minimal live requirement $(\bar{s}, \gamma)$. That is, for every live requirement (s', β) in CH_k, $depth_k(\bar{s}) \leq depth_k(s')$.

Case $\gamma \equiv \Diamond\alpha$: We have a future requirement $(\bar{s}, \Diamond\alpha)$. Since $\Diamond\alpha \in T(\bar{s})$, there exists an MCS B such that $\{\beta \mid \Box\beta \in T(\bar{s})\} \cup \{\alpha\} \subseteq B$, thanks to Proposition 4.6. Since TS_k is finite, $S_k \subset \widehat{S}$. Pick $\widehat{s} \in \widehat{S} - S_k$. Define $S_{k+1} \stackrel{\text{def}}{=} S_k \cup \{\widehat{s}\}$. Further $\rightarrow_k$ is finite, hence $\Sigma_k \stackrel{\text{def}}{=} Voc \cup \{a \mid \exists s_1, s_2 \in S_k : s_1 \xrightarrow{a}_k s_2\}$ is finite. Thus $\Sigma_k \subset \Sigma$. Pick $d \in \Sigma - \Sigma_k$. Define

$$\rightarrow_{k+1} \stackrel{\text{def}}{=} \rightarrow_k \cup \{(\bar{s}, \{d\}, \widehat{s}), (\widehat{s}, \emptyset, \widehat{s})\}.$$

Extend T_k to T_{k+1} by setting $T_{k+1}(\widehat{s}) = B$. Now $TS_{k+1} = (S_{k+1}, \rightarrow_{k+1})$ and $CH_{k+1} = (TS_{k+1}, T_{k+1})$. It is easy to check that CH_{k+1} is a chronicle structure satisfying conditions (A), (B) and (C).

Case $\gamma \equiv \langle\bar{a}\rangle True$: We have a successor requirement $(\bar{s}, \langle\bar{a}\rangle True)$ which is $depth_k$-minimal. Since $\langle\bar{a}\rangle True \in T_k(\bar{s})$, by Proposition 4.7, there exists an MCS B such that $\{\beta \mid [\bar{a}]\beta \in A\} \subseteq B$. Again since TS_k is finite, we can pick $\widehat{s} \in \widehat{S} - S_k$ and let $S_{k+1} \stackrel{\text{def}}{=} S_k \cup \{\widehat{s}\}$. Define

$$\rightarrow_{k+1} \stackrel{\text{def}}{=} \rightarrow_k \cup \{(\widehat{s}, \emptyset, \widehat{s})\} \cup \{(s, b, \widehat{s}) \mid (s, b) =_k (\bar{s}, \bar{a})\}$$

Extend T_k to T_{k+1} by letting $T_{k+1}(\widehat{s}) = B$. Now $TS_{k+1} = (S_{k+1}, \rightarrow_{k+1})$ and $CH_{k+1} = (TS_{k+1}, T_{k+1})$. We now show that CH_{k+1} is a chronicle structure satisfying conditions (A), (B) and (C).

Clearly TS_{k+1} is a finite, sequential dts. Further the restriction of $\rightarrow_{k+1}$ and T_{k+1} to S yields $\rightarrow_k$ and T_k, as required. Note that $\mathcal{R}_{TS_{k+1}}(\widehat{s}) = \{\widehat{s}\}$. Since $S_{k+1} - S_k = \{\widehat{s}\}$, we thus see that TS_{k+1} is acyclic. Since $\widehat{s} \in \mathcal{R}_{TS_{k+1}}(\widehat{s_0})$, TS_{k+1} is pointed with root $\widehat{s_0}$. Determinacy of TS_{k+1} follows from the observation that $s \xrightarrow{b}_{k+1} \widehat{s}$ only when $(s, \langle b\rangle True)$ is a live successor requirement in CH_k.

To show that TS_{k+1} is graded, let $s \xrightarrow{b}_{k+1} s'$. If s and s' are both in S_k we are done since TS_k is graded. By the earlier observation, we know that $s \neq \widehat{s}$. Thus let $s \in S_k$ and $s' = \widehat{s}$. We have $s \xrightarrow{b}_{k+1} \widehat{s}$ and need to show that $depth_{k+1}(\widehat{s}) = 1 + depth_{k+1}(s) = 1 + depth_k(s)$. Now, by definition of the depth function, $depth_{k+1}(\widehat{s}) \geq 1 + depth_{k+1}(s)$. So suppose $depth_{k+1}(\widehat{s}) > 1 + depth_k(s)$. Then there exists $s' \in S_{k+1}$ and $a \in \Sigma$ such that $s' \xrightarrow{a}_{k+1} \widehat{s}$ and $depth_{k+1}(\widehat{s}) = 1 + depth_{k+1}(s')$. By construction, we must have $s' \in S_k$ and $(s', a) =_k (\bar{s}, \bar{a}) =_k (s, b)$ and hence $depth_k(s') = depth_k(s)$ giving a contradiction.

We have shown that CH_{k+1} indeed satisfies conditions (A) and (C). To show α_0-coherence of T_{k+1}, consider $b \in Voc$ such that $s \xrightarrow{b}_{k+1} s'$ and $[b]\alpha \in T_{k+1}(s)$. We need to show that $\alpha \in T_{k+1}(s')$.

It suffices to consider the case when $s \in S_k$ and $s' = \widehat{s}$. We have $(s, b) =_k (\bar{s}, \bar{a})$. Let $s_1, s_2, \ldots, s_{j-1} \in S_k$ and $a_1, a_2, \ldots a_{j-1} \in Voc$ such that

$$(\bar{s}, \bar{a}) = (s_0, a_0) \doteq_k (s_1, a_1) \doteq_k \ldots \doteq_k (s_{j-1}, a_{j-1}) \doteq_k (s_j, a_j) = (s, b).$$

We show by induction on i that $\{\beta \mid [a_i]\beta \in T_k(s_i)\} \subseteq T_{k+1}(\widehat{s}) = B$. The base case, when $i = 0$, comes about by choice of B. Let $i > 0$ and suppose that $[a_i]\beta \in T_k(s_i)$. Let $s'' \in S_k$ such that $\langle\{a_{i-1}, a_i\}\rangle True \in T_k(s'')$ and $s'' \xrightarrow{a_i}_k s_{i-1}, s'' \xrightarrow{a_{i-1}}_k s_i$. The existence of such an s'' is guaranteed by the third condition in the definition of $\doteq_k$. By α_0-coherence of T_k, $\langle a_{i-1}\rangle[a_i]\beta \in T_k(s'')$. By thesis $(T8)$, $[\{a_{i-1}, a_i\}]\beta \in T_k(s'')$. Since $\langle\{a_{i-1}, a_i\}\rangle True \in T_k(s'')$, by thesis $(T9)$, $[a_i][a_{i-1}]\beta \in T_k(s'')$. By α_0-coherence of T_k, $[a_{i-1}]\beta \in T_k(s_{i-1})$. By the induction hypothesis on i, $\beta \in B$, as required.

Further, if $\Box\alpha \in T_k(s)$, by thesis $(T6)$, $[b]\Box\alpha \in T_k(s)$. We have just shown that in that case, $\Box\alpha \in B$ and by thesis $(T5)$, $\alpha \in B$ as well. Thus T_{k+1} is α_0-coherent and the inductive construction of CH_{k+1} is complete.

Define $CH \stackrel{\text{def}}{=} (TS, T)$ where $TS \stackrel{\text{def}}{=} (S, \rightarrow)$ by

$$S \stackrel{\text{def}}{=} \bigcup_{k \geq 0} S_k, \quad \rightarrow \stackrel{\text{def}}{=} \bigcup_{k \geq 0} \rightarrow_k, \quad \text{and } T(s) = T_k(s), \text{ for } s \in S_k.$$

T is well-defined since $T_k = T_{k+1} \lceil S_k$, for all k. Notice that a "fresh" action d outside Voc is used to satisfy all future requirements and that once a successor requirement $(s, \langle a\rangle True)$ is satisfied, no further a transitions can be added. Hence TS is a deterministic dts. It is easy to verify that T is α_0-coherent.

Towards showing that CH is α_0-perfect, observe the following:

Claim 1: Let $u \subseteq Voc$. If $s, s_1, s_2 \in S$ such that there are thin u-cubes f and g respectively from s to s_1 and s to s_2, then for every $v \subseteq u$, $f(v) = g(v)$.

The proof is by induction on u, using determinacy of TS.

Claim 2: For every $s \in S$ and $a \in Voc$, if $\langle a\rangle True \in T(s)$ then there is an $s' \in S$ such that $s \xrightarrow{a} s'$.

Let $k = min\{j \mid s \in S_j\}$. Note that for every $s' \in S - S_k$, $depth_{TS}(s') \geq depth_{TS}(s)$. Since TS_k is finite, let $m = |\{s' \in S_k \mid s' \neq s, depth_k(s') < depth(s)\}|$. Either $(s, \langle a\rangle True)$ is not a live successor requirement in CH_{k+m} (in which case we are done) or it is a $depth_{k+m}$-minimal successor requirement in TS_{k+m}. Let there be n such minimal requirements in CH_{k+m}. Surely none of them can be live in CH_{k+m+n} and we are done.

Claim 3: Let $s \in S, u \subseteq Voc$ and $\langle u\rangle True \in T(s)$. Then there exists $s' \in S$ such that there is a thin u-cube from s to s'.

The proof is by induction on $|u|$. The base case, when $u = \emptyset$ is trivial. For the induction step, let $u = \{a_1, \ldots, a_n\}$. By the induction hypothesis we can assume, for every $i \in \{1, \ldots, n\}$, an $s_i \in S_k$ with a thin $(u - \{a_i\})$-cube f_i from s to s_i. Since $\langle u \rangle True \in T(s)$, by thesis ($T8$), for every i, $[u - \{a_i\}]\langle a_i \rangle True \in T(s)$. By α_0-coherence of T and the fact that $\langle u - \{a_i\}\rangle True \in T(s)$ (using $T7$), Proposition 4.4 assures us that $\langle a_i \rangle True \in T(s_i)$. By the previous claim, for every i, there exists s'_i such that $s_i \xrightarrow{a_i} s'_i$.

Let k be one less than the least j such that one of the $s'_i \in S_j$. Clearly, for all i, $(s_i, \langle a_i \rangle True)$ is a live successor requirement in CH_k. Now consider two thin cubes, f_i a thin $(u - \{a_i\})$-cube to s_i and f_j a thin $(u - \{a_j\})$-cube to s_j, $i \neq j$. Let $v = u - \{a_i, a_j\}$. By Claim 1, $f_i(v) = f_j(v) = s''$ (say). Now f_i defines a thin v-cube from s to s''. Further since $\langle u \rangle True \in T(s)$, by thesis ($T8$), $[v]\langle\{a_i, a_j\}\rangle True \in T(s)$ and by α_0-coherence of T and Proposition 4.4, $\langle\{a_i, a_j\}\rangle True \in T(s'')$. Further, $s'' \xrightarrow{a_j} s_i$ and $s'' \xrightarrow{a_i} s_j$. Hence $(s_i, a_i) \doteq_k (s_j, a_j)$. Since we know that $s'_i \in S_{k+1} - S_k$, the chosen live successor requirement at stage k must be equivalent to (s_i, a_i) and hence (s_j, a_j). By construction, for every i, $s_i \xrightarrow{a_i}_{k+1} \widehat{s} = s'_i$. We now define the thin u-cube f from s to $\widehat{s}$ by:

$$f(v) \stackrel{\text{def}}{=} \begin{cases} f_i(v), & v \subseteq u - \{a_i\}, a_i \in u \\ \widehat{s}, & v = u. \end{cases}$$

It can be easily shown that CH is α_0-perfect. $\square$

Theorem 4.9 (Completeness) *If* $\models_{Det} \alpha$ *then* $\vdash_D \alpha$.

5 Undecidability

In this section and the subsequent sections, our emphasis will be on negative results. Specifically, we shall show that the satisfiability problem for our logic becomes undecidable when some natural restrictions are placed on the class of permissible models.

We first consider *deterministic* distributed transition systems over a countably infinite alphabet Σ. We begin by showing that deterministic satisfiability is undecidable, or in other words, that the set $DSAT$ is not recursive.

Various versions of the *colouring problem* [Parikh] will be used to establish our negative results. Colouring problems correspond to *tiling problems* (see [Har85]) and in this section the colouring problem that we consider (called simply CP) corresponds to the so-called origin constrained tiling problem in [Har85].

An *instance* of CP is a triple $\Delta = (C, R, U)$ where $C = \{c_0, c_1, \ldots, c_k\}$ is a finite non-empty set of colours and $R, U : C \to (\wp(C) - \emptyset)$ are the "right" and "up" functions.

A *solution* to Δ is a colouring function $Col : \mathbf{N} \times \mathbf{N} \to C$ which satisfies:

1. $Col(0, 0) = c_0$.
2. $\forall (i, j) \in \mathbf{N} \times \mathbf{N}$, $Col(i+1, j) \in R(Col(i, j))$ and $Col(i, j+1) \in U(Col(i, j))$.

It follows easily from [Har85] as shown in [Parikh] that CP is Σ^0_1-complete and hence undecidable.

We now reduce each instance of CP to a membership problem for $DSAT$. In other words, we shall uniformly encode each instance Δ of CP into a formula α_Δ such that Δ has a solution iff $\alpha_\Delta \in DSAT$. In order to capture the effects of functions R and U, we reserve two actions x and y respectively in Σ. We reserve $k+1$ atomic propositions in P to denote the colours in $C = \{c_0, c_1, \ldots, c_k\}$. For notational convenience, these atomic propositions will also be written as $c_0, c_1, \ldots, c_k$.

Definition 5.1 *Let $\Delta = (C, R, U)$ be an instance of CP, where C is a nonempty finite set $\{c_0, \ldots, c_k\}$. Then $\alpha_\Delta \stackrel{\text{def}}{=} \bigwedge_{i=1}^{5} \alpha_i$, where*

- $\alpha_1 \stackrel{\text{def}}{=} c_0$.
- $\alpha_2 \stackrel{\text{def}}{=} \Box\langle\{x,y\}\rangle True$.
- $\alpha_3 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \equiv \bigwedge_{j\neq i} {\sim} c_j)$.
- $\alpha_4 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \supset [x] \bigvee_{c\in R(c_i)} c)$.
- $\alpha_5 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \supset [y] \bigvee_{c\in U(c_i)} c)$.

The intended meaning of the conjuncts of α_Δ should be clear. The important formula is α_2 which, in the presence of determinacy, encodes the "grid" $\mathbf{N}\times\mathbf{N}$ (as we saw in Figure 8).

Lemma 5.2 *Let $\Delta = (C, R, U)$ be an instance of CP. If Δ has a solution, then $\alpha_\Delta \in DSAT$.*

Proof: Let $Col : \mathbf{N}\times\mathbf{N} \to C$ be a solution to CP. Define now $TS = (S, \to)$ as follows:

$$\begin{array}{lcl} S & \stackrel{\text{def}}{=} & \mathbf{N}\times\mathbf{N}. \\ \to & \stackrel{\text{def}}{=} & \{((i,j),\{x\},(i+1,j)) \mid (i,j)\in\mathbf{N}\times\mathbf{N}\} \\ & & \cup\{((i,j),\{y\},(i,j+1)) \mid (i,j)\in\mathbf{N}\times\mathbf{N}\} \\ & & \cup\{((i,j),\{x,y\},(i+1,j+1)) \mid (i,j)\in\mathbf{N}\times\mathbf{N}\} \\ & & \cup\{((i,j),\emptyset,(i,j)) \mid (i,j)\in\mathbf{N}\times\mathbf{N}\} \end{array}$$

Then it is clear that TS is a deterministic dts over Σ. Next define $V : S \to \wp(P)$ as: $V(i,j) \stackrel{\text{def}}{=} \{Col(i,j)\}$. Let $M = (TS, V)$. Then it is straightforward to show that $M, (0,0) \models \alpha_\Delta$. $\Box$

The converse of this lemma is more difficult to prove. We first prove an intermediate result.

Lemma 5.3 *Let $M = (TS, V)$ be a model where $TS = (S, \rightarrow)$ is a deterministic dts over Σ. Let $s \in S$ such that $M, s \models \langle\{x, y\}\rangle True$ and let $s' \in S$. Then the following statements are equivalent:*

1. $s \xrightarrow{\{x,y\}} s'$.
2. $\exists s_x \in S : s \xrightarrow{x} s_x \xrightarrow{y} s'$.
3. $\exists s_y \in S : s \xrightarrow{y} s_y \xrightarrow{x} s'$.

Proof: From the definition of a dts, it follows that (1) implies (2) and (3). So now suppose that (2) holds. Since $M, s \models \langle\{x, y\}\rangle True$, there exists $s'' \in S$ such that $s \xrightarrow{\{x,y\}} s''$. Hence, for some $s_1 \in S$, we have $s \xrightarrow{x} s_1 \xrightarrow{y} s''$. But TS is deterministic, hence $s_1 = s_x$. But then $s_x \xrightarrow{y} s'$ and $s_x \xrightarrow{y} s''$ and again by determinacy of TS, we get $s' = s''$. Therefore (1) holds. Similarly we can show that (3) implies (1). $\square$

Lemma 5.4 *Let $\Delta = (C, R, U)$ be an instance of CP such that $\alpha_\Delta \in DSAT$. Then Δ has a solution.*

Proof: Let $M, s_0 \models \alpha_\Delta$, where $M = (TS, V)$, $TS = (S, \rightarrow)$ is a deterministic dts over Σ and $s_0 \in S$.

Towards constructing a colouring function for Δ, we adapt the following strategy: we first compute the colours on the diagonal in $\mathbf{N} \times \mathbf{N}$ and then inductively fill out larger and larger squares. For each point on the grid, we associate a state in $\mathcal{R}(s_0)$; this is sufficient since the formula $\alpha_3 \wedge \alpha_4 \wedge \alpha_5$ is satisfied at that state and hence the colouring function can be easily "pulled out".

The diagonal function $Diag : \mathbf{N} \rightarrow \mathcal{R}(s_0)$ is defined inductively:

- $Diag(0) \stackrel{\text{def}}{=} s_0$.
- $Diag(m+1) \stackrel{\text{def}}{=} s$, provided $Diag(m) \xrightarrow{\{x,y\}} s$.

Since $M, s_0 \models \alpha_2$, for every $s \in \mathcal{R}(s_0)$, s has an $\{x, y\}$-successor and hence $Diag$ is total. Determinacy of TS ensures that $Diag$ is well-defined. We have $Diag(i) \xrightarrow{\{x,y\}} Diag(i+1)$, for all i, directly from the definition.

In what follows, let i, j, m and n range over $\mathbf{N}$. We now construct a sequence of function pairs $\{(\Psi_m, Col_m)\}_{m \geq 0}$ with $\Psi_m : \{0, \ldots, m\} \times \{0, \ldots, m\} \rightarrow S$ and $Col_m : \{0, \ldots, m\} \times \{0, \ldots, m\} \rightarrow C$ such that the following conditions are satisfied at every stage $m, m \geq 0$:

$$\begin{array}{lll}
(C1) & Col_m(0,0) = c_0 & \\
(C2) & \Psi_m(i,j) \xrightarrow{x} \Psi_m(i+1,j) & [0 \leq i < m, 0 \leq j \leq m] \\
(C3) & \Psi_m(i,j) \xrightarrow{y} \Psi_m(i,j+1) & [0 \leq i \leq m, 0 \leq j < m] \\
(C4) & \Psi_m(i,i) = Diag(i) & [0 \leq i \leq m] \\
(C5) & Col_m(i+1,j) \in R(Col_m(i,j)) & [0 \leq i < m, 0 \leq j \leq m] \\
(C6) & Col_m(i,j+1) \in U(Col_m(i,j)) & [0 \leq i \leq m, 0 \leq j < m]
\end{array}$$

Set $\Psi_0(0,0) \stackrel{\text{def}}{=} s_0$ and $Col_0(0,0) \stackrel{\text{def}}{=} c_0$. Clearly conditions C1 and C4 are satisfied and the rest of the conditions are satisfied vacuously.

Assume that inductively we are given Ψ_m, Col_m. Ψ_{m+1}, Col_{m+1} are now defined, in five steps:

Step 1: Set

$$\Psi_{m+1}(i,j) \stackrel{\text{def}}{=} \Psi_m(i,j)\ [0 \le i \le m,\ 0 \le j \le m]\ \textit{and}$$
$$Col_{m+1}(i,j) \stackrel{\text{def}}{=} Col_m(i,j)\ [0 \le i \le m,\ 0 \le j \le m].$$

This ensures that Ψ_{m+1} restricted to $\{0,\ldots,m\} \times \{0,\ldots,m\}$ is Ψ_m and a similar statement holds for Col_m. Further, this guarantees that Col_{m+1} satisfies condition C1.

Step 2: Set $\Psi_{m+1}(m+1,m+1) \stackrel{\text{def}}{=} Diag(m+1)$. This ensures C4 for Ψ_{m+1} and that $\Psi_{m+1}(m,m) \xrightarrow{\{x,y\}} \Psi_{m+1}(m+1,m+1)$.

Step 3: We now define $\Psi_{m+1}(m+1,j)$, for $0 \le j \le m$, by induction on $m-j$. For the base case, we have $j = m$. We have $\Psi_{m+1}(m,m) \xrightarrow{\{x,y\}} \Psi_{m+1}(m+1,m+1)$ by Step 2. Hence there exists $s_y \in \mathcal{R}(s_0)$ such that $\Psi_{m+1}(m,m) \xrightarrow{y} s_y \xrightarrow{x} \Psi_{m+1}(m+1,m+1)$. By determinacy of TS, s_y is unique. Define $\Psi_{m+1}(m+1,m) \stackrel{\text{def}}{=} s_y$. Since Ψ_m satisfies C2, by Step 1, we have $\Psi_{m+1}(m,m-1) \xrightarrow{x} \Psi_{m+1}(m,m)$. Now, by Lemma 5.3, we get $\Psi_{m+1}(m,m-1) \xrightarrow{\{x,y\}} \Psi_{m+1}(m+1,m)$.

For the inductive step, we have $j < m$. By induction hypothesis, we can assume $\Psi_{m+1}(m,j) \xrightarrow{\{x,y\}} \Psi_{m+1}(m+1,j+1)$. By similar reasoning as above, we determine $\Psi_{m+1}(m+1,j)$. Thus, the $(m+1)^{th}$ row is completely defined, and Ψ_{m+1} satisfies condition C2.

Step 4: The definition of $\Psi_{m+1}(i,m+1)$, for $[0 \le i \le m]$ proceeds in the same manner as in Step 3, except that we now appeal to the fact that Ψ_m satisfies C3 and inductively ensure that $\Psi_{m+1}(i,m) \xrightarrow{\{x,y\}} \Psi_{m+1}(i+1,m+1)$. Thus, the $(m+1)^{th}$ column is completely defined, and Ψ_{m+1} satisfies condition C3.

Step 5: We now define $Col_{m+1}(i,j)$, for $i > m$ or $j > m$ to be simply the colour c, where $c \in V(\Psi_{m+1}(i,j))$. Since $\Psi_{m+1}(i,j) \in \mathcal{R}(s_0)$, α_3, α_4 and α_5 ensure that $Col_{m+1}(i,j)$ is well-defined for these values and that Col_{m+1} satisfies conditions C5 and C6.

This completes the inductive construction of Ψ_{m+1} and Col_{m+1}. Finally define $Col : \mathbf{N} \times \mathbf{N} \to C$ by $Col(i,j) \stackrel{\text{def}}{=} Col_m(i,j)$, where $m = max\{i,j\}$. It is easy to verify that Col is a solution to Δ. □

Theorem 5.5 *Deterministic satisfiability is undecidable.*

Proof: By the earlier Lemma 5.2 and Lemma 5.4, any instance Δ of CP has a solution iff the formula $\alpha_\Delta \in DSAT$. Since CP is undecidable, so is membership in $DSAT$. □

Actually, the proof of Lemma 5.4 is more elaborate than necessary. We have chosen this method to emphasize that it is not determinacy as such, but a weaker property implied by determinacy which yields undecidability. This property is specified in Lemma 5.3 and it can arise in a natural way even in the absence of determinacy. In particular, the partial commutativity of actions, as it occurs in the theory of *trace languages*, gives rise to the same phenomenon. The reader can verify that the undecidability proof goes through for a (possibly nondeterministic) dts $TS = (S, \Sigma, \rightarrow)$ which satisfies, for some $a, b \in \Sigma$,

for every $s_0, s_1, s_2 \in S$, if $s_0 \xrightarrow{a} s_1 \xrightarrow{b} s_2$ then $s_0 \xrightarrow{\{a,b\}} s_2$.

Such transition systems occur in the theory of trace languages [Maz] and we shall show in Section 8 how the satisfiability problem for an appropriate logical language is undecidable.

6 DTS's over Finite Alphabets

So far we have considered dts's over Σ, where Σ is a countably infinite alphabet set. We now turn to a natural variant, namely the class of dts's over finite alphabets.

Due to the mixture of temporal and step operators in our logical language Step-TL, there is a significant difference between the finite and infinite alphabet cases. This is so because formulas of the form $\Diamond\alpha$ can be more easily satisfied when the alphabet is infinite. (The same observation holds for any action-indexed temporal logic.)

We first introduce some useful terminology for the finite case. For convenience, we consider only finite nonempty subsets of Σ as our finite alphabets.

Let A be a finite nonempty subset of Σ. A dts *over* A is a dts $(TS = (S, \rightarrow)$ such that $\rightarrow \subseteq S \times \wp(A) \times S$. An *$A$-frame* is a dts over A. An *A-model* is a model $M = (TS, V)$, where TS is an A-frame. $\{\alpha \mid Voc(\alpha) \subseteq A\}$ is the set of *A-formulas.* The A-formula α is *A-satisfiable* iff there exists an A-model $M = ((S, \rightarrow), V)$ and $s \in S$ such that $M, s \models \alpha$. The notion of A-validity (restricted to A-formulas) is defined in the obvious way. We write $\models_A \alpha$ to denote that the formula α is A-valid.

Now the formula $\beta \stackrel{\text{def}}{=} p \wedge \Diamond{\sim}p \wedge \bigwedge_{a \in A} [a]False$ is obviously not A-satisfiable, but is certainly (Σ-)satisfiable. This is the essence of the difference between the finite and infinite alphabet cases. The relationship between the two notions of satisfiability can be brought out as a corollary of the completeness theorem (Theorem 2.7) in Section 2:

Corollary 6.1 *α is satisfiable iff α is A-satisfiable for some $A \in \wp_{fin}(\Sigma)$ with $Voc(\alpha) \subseteq A$ and $|A| \leq |Voc(\alpha)| + 1$.*

Proof: Suppose α is satisfiable. Then by the soundness theorem for ND, α is ND-consistent. By the proof of Theorem 2.7, α is A-satisfiable for some $A \in \wp_{fin}(\Sigma)$ with $Voc(\alpha) \subseteq A$ and $|A| \leq |Voc(\alpha)| + 1$.

The second half of the result is immediate because every dts over A is also a dts over Σ. $\square$

For the rest of this section, we fix A, a finite nonempty subset of Σ. Our first aim is to consider the set of A-valid formulas. Let ND^0_A denote the axiom system ND (presented in Section 2) *instantiated over A-formulas.* It is easy to see that ND_A^0 is sound over the class of A-models, but it cannot be complete. This is because the formula β defined above is ND_A^0-consistent (by soundness of ND, since β is satisfiable), but not A-satisfiable. For completeness, we need in addition the following *induction scheme*:

$$(AIndn) \quad \Box(\alpha \supset \bigwedge_{a \in A} [a]\alpha) \supset (\alpha \supset \Box\alpha)$$

Let ND_A stand for the system ND_A^0 augmented with $(AIndn)$. The following theorem can be easily proved:

Theorem 6.2 (Soundness) *If α is a thesis derivable from ND_A then α is valid over the class of A-models.*

The completeness proof proceeds along the lines of the proof of Theorem 2.7. We assume the notation and terminology of that proof for the discussion below. Let α_0 be an ND_A-consistent formula. We first define $TS_0 = (AT, \rightarrow)$ by:

$$w \xrightarrow{u} w' \text{ iff } \widehat{w} \wedge \langle u \rangle \widehat{w'} \text{ is consistent}, \; u \subseteq A$$

It can be easily checked, as in the proof of Lemma 2.4, that TS_0 is a dts. Clearly, TS_0 is an A-dts.

The rest of the proof proceeds exactly as before, the only difference being that $M_0, w \models \Diamond\alpha$ when $\Diamond\alpha \in w$. To establish this, we need an intermediate result:

Lemma 6.3 *Let $w \in AT$ and $R = \mathcal{R}_{TS_0}(w)$. Then $\vdash \widetilde{R} \supset \bigwedge_{a \in A} [a]\widetilde{R}$.*

Proof: Assume w and R as above. If $R = AT$, then from $\vdash \widetilde{AT}$ (Proposition 2.3(iv)) and TG, we get $\bigwedge_{a \in A} [a]\widetilde{R}$, and hence, by PC, the formula above.

Otherwise let $R = \{x_1, \ldots, x_k\}$ and let $AT - R = \{y_1, \ldots, y_l\}$. Suppose the formula is not a thesis. Then $\widetilde{R} \wedge \bigvee_{a \in A} \langle a \rangle {\sim}\widetilde{R}$ is consistent. By Proposition 2.3(iv), $\vdash \widetilde{AT}$ and hence we can show that $\vdash {\sim}\widetilde{R} \supset \widehat{y_1} \vee \ldots \vee \widehat{y_l}$. Thus, $(\widehat{x_1} \vee \ldots \vee \widehat{x_k}) \wedge \bigvee_{a \in A} \langle a \rangle (\widehat{y_1} \vee \ldots \vee \widehat{y_l})$ is consistent. Hence, for some $i \in \{1, \ldots, k\}$, some $j \in \{1, \ldots, l\}$, and some $a \in A$, $\widehat{x_i} \wedge \langle a \rangle \widehat{y_j}$ is consistent. By definition of $\rightarrow$, we get $x_i \xrightarrow{a} y_j$. But then $x_i \in \mathcal{R}_{TS_0}(w)$ and hence $y_j \in \mathcal{R}_{TS_0}(w)$ as well, contradicting our assumption that $y_j \in AT - R$. $\Box$

Lemma 6.4 *Let $w \in AT$ and let $\Diamond\alpha \in w$. Then, for some $w' \in \mathcal{R}(w), \alpha \in w'$.*

Proof: Suppose $\Diamond\alpha \in w$. Let $R = \mathcal{R}_{TS_0}(w)$. By the Lemma above, $\vdash \widetilde{R} \supset \bigwedge_{a \in A} [a]\widetilde{R}$. By the rule (TG), we get $\vdash \Box(\widetilde{R} \supset \bigwedge_{a \in A} [a]\widetilde{R})$. By the axiom $(AIndn)$, we get $\vdash \widetilde{R} \supset \Box\widetilde{R}$. Since $w \in R$, we have

$\vdash \widehat{w} \supset \widetilde{R}$ and hence, $\vdash \widehat{w} \supset \Box\widetilde{R}$. Since $\Diamond\alpha \in w$, $\vdash \widehat{w} \supset \Box\widetilde{R} \wedge \Diamond\alpha$. Hence $\vdash \widehat{w} \supset \Diamond(\widetilde{R} \wedge \alpha)$. Using the rule (TG), we find that $(\widetilde{R} \wedge \alpha)$ is consistent. Hence there exists $w' \in R$ such that $\widehat{w'} \wedge \alpha$ is consistent. That is, $\alpha \in w'$ and the lemma is proved. □

The remaining details are as in Section 2. We then get:

Theorem 6.5 (Completeness and decidability)

1. *For any A-formula α, if $\models_A \alpha$, then $\vdash_{ND_A} \alpha$.*
2. *A-satisfiability is decidable in nondeterministic exponential time.*

It is straightforward to establish the results of Section 3 on prime event structures and net systems with minor notational modifications for A-formulas.

We now turn to *deterministic* dts's over a finite alphabet $A \subseteq \Sigma$. We can then define $DSAT_A$ and $DVAL_A$ in the obvious way. The case when $|A| = 1$ is standard: decidability can be proved and an axiomatization found (see, for example, [Gol]). For $|A| > 1$, from the results of the previous section, it is clear that $DSAT_A$ is not a recursive set. The main surprise is that $DVAL_A$ is *not* recursively enumerable either! Hence the completeness argument given in Section 4 cannot go through. (There, we managed to build a deterministic model for a consistent formula by picking a *new* element from Σ to satisfy each future requirement. We cannot do this when the alphabet is finite.) We will prove that $DVAL_A$, the set of all A-formulas valid over the class of deterministic A-models, is Π^1_1-complete and hence not axiomatizable.

We use the so-called *Recurring Colouring Problem* (RCP) to obtain our negative result. As one may expect, RCP is recursively equivalent to the Recurring Tiling Problem considered in [Har85] and the equivalence between the two problems is shown in [Parikh].

An *instance* of RCP is a tuple $\Delta = (C, R, U, c_r)$ where $C = \{c_0, c_1, \ldots, c_k\}$ is a finite non-empty set of colours, $c_r \in C$ and $R, U : C \to (\wp(C) - \emptyset)$ are the "right" and "up" functions.

A *solution* to Δ is a colouring function $Col : \mathbf{N} \times \mathbf{N} \to C$ which satisfies:

1. $Col(0,0) = c_0$.
2. $\forall (i,j) \in \mathbf{N} \times \mathbf{N},\ Col(i+1,j) \in R(Col(i,j))$ and $Col(i,j+1) \in U(Col(i,j))$.
3. $\forall m \in \mathbf{N} :\ \exists n > m :\ Col(0,n) = c_r$.

Thus RCP is CP with an additional constraint, which can alternatively be stated as: along the Y-axis, an infinite number of grid points are to be coloured with the recurring colour c_r.

We shall encode each instance Δ of RCP into an A-formula β_Δ and prove that Δ has a solution iff $\beta_\Delta \in DSAT_A$. For coding the functions R and U, we reserve two letters x and y as before (this is why we need $|A| > 1$). For convenience, we again assume $C \subseteq P$. In addition, we reserve five atomic propositions (disjoint from C) denoted $\{Y, D, AD, BD, RR\}$. Y will be used to mark the points lying on the Y-axis. D will be used to mark the diagonal line of the grid. BD and AD

respectively will be used to mark the points below and above the diagonal. Finally, RR will be used to pick out the lines parallel to the X-axis, whose intersections with the Y-axis have been assigned the recurring colour c_r. (Actually, for proving the negative result for $DSAT_A$, we do not need the last four special propositions; we introduce them only so that a uniform proof can be given for trace transition systems to be introduced in Section 8.)

Definition 6.6 *Let* $\Delta = (C, R, U, c_r)$ *be an instance of RCP, where* $C = \{c_0, c_1, \ldots, c_k\}$ *and* $c_r \in C$. *Then,* $\beta_\Delta \stackrel{\text{def}}{=} \bigwedge_{i=1}^{10} \beta_i$, *where*

1. $\beta_1 \stackrel{\text{def}}{=} c_0$.
2. $\beta_2 \stackrel{\text{def}}{=} \Box(\langle\{x,y\}\rangle True \wedge \bigwedge_{a \notin \{x,y\}} [a]False)$.
3. $\beta_3 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \equiv \bigwedge_{j \neq i} \sim c_j)$.
4. $\beta_4 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \supset [x] \bigvee_{c \in R(c_i)} c)$.
5. $\beta_5 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \supset [y] \bigvee_{c \in U(c_i)} c)$.
6. $\beta_6 \stackrel{\text{def}}{=} \Box((D \wedge \sim BD \wedge \sim AD) \vee (BD \wedge \sim D \wedge \sim AD) \vee (AD \wedge \sim D \wedge \sim BD))$
7. $\beta_7 \stackrel{\text{def}}{=} D \wedge \Box(D \supset ([\{x,y\}]D \wedge [x]BD \wedge [y]AD \wedge \Diamond(D \wedge RR)))$
8. $\beta_8 \stackrel{\text{def}}{=} \Box(BD \supset [x]BD) \wedge \Box(AD \supset [y]AD)$
9. $\beta_9 \stackrel{\text{def}}{=} \Box((\langle x\rangle RR \supset RR) \wedge (RR \supset [x]RR))$
10. $\beta_{10} \stackrel{\text{def}}{=} Y \wedge \Box(Y \supset ([y]Y \wedge [x]\Box \sim Y \wedge (RR \supset c_r)))$

β_1 through β_5 are just like α_1 through α_5 in the definition of α_Δ in Section 5, except that β_2 is a strengthened version, where we exploit the fact that A is finite, and force models satisfying β_Δ to be based on $\{x,y\}$-frames. This turns out to be crucial in enforcing the recurrence constraint along the Y-axis. β_6 to β_8 describe the diagonal points, and the ones below and above them. Further, β_7 ensures that an infinite number of diagonal points are marked by RR as belonging to the recurrence row. β_9 propagates the recurrence row information along the x-direction to the right and the left. β_{10} describes the Y-axis and ensures that points lying on its intersection with the recurrence rows are coloured by c_r.

Before we present the proof of the reduction, let us introduce some notation to extend the transition relation to sequences of actions; this will be useful through this and the next section of the paper. For a dts $(S, \Sigma, \rightarrow)$, we define the transition relation $\Rightarrow \subseteq S \times \Sigma^* \times S$ inductively by:

- $s \overset{\lambda}{\Rightarrow} s$ for every $s \in S$. (Here λ denotes the null string.)
- If $s \overset{\rho}{\Rightarrow} s'$ and $s' \overset{a}{\rightarrow} s''$, $a \in \Sigma$ then $s \overset{\rho a}{\Longrightarrow} s''$.

Secondly, for $\rho \in \Sigma^*$ and $n \in \mathbf{N}$, the string ρ^n is given inductively by:

- $\rho^0 = \lambda$.
- $\rho^{n+1} = \rho^n \rho$.

Finally, for $\rho \in \Sigma^*$ and $a \in \Sigma$, let $\sharp_a(\rho)$ denote the number of occurrences of the symbol a in the sequence ρ.

Lemma 6.7 *Let $\Delta = (C, R, U, c_r)$ be an instance of RCP. If Δ has a solution, then $\beta_\Delta \in DSAT_A$.*

Proof: Let $Col : \mathbf{N} \times \mathbf{N} \to C$ be a solution to Δ. Define now $TS = (S, \rightarrow)$ as in Lemma 5.2. Then it is clear that TS is a deterministic A-dts. Next define $V : S \to \wp(P)$ to be a function which satisfies, for all $i, j \in \mathbf{N}$:

1. $V(i,j) \subseteq \{Col(i,j)\} \cup \{D, BD, AD, Y, RR\}$.
2. $D \in V(i,j)$ iff $i = j$; $BD \in V(i,j)$ iff $i < j$; $AD \in V(i,j)$ iff $i > j$.
3. $RR \in V(i,j)$ iff $Col(0,j) = c_r$.
4. $Y \in V(i,j)$ iff $i = 0$.

Clearly, V is a well-defined map. Let $M = (TS, V)$. Then it is straightforward to show that $M, (0,0) \models \beta_\Delta$. □

To prove the converse, we need some intermediate results. Firstly recall that Lemma 5.3 showed that in a deterministic model, when $\langle\{a,b\}\rangle True$ holds at a state s, we have $s \overset{\{a,b\}}{\longrightarrow} s'$ iff $s \overset{ab}{\Longrightarrow} s'$ iff $s \overset{ba}{\Longrightarrow} s'$. This result, of course, holds for deterministic A-frames as well.

Lemma 6.8 *Let Δ be an instance of RCP and $M = ((S, \rightarrow), V)$ be a deterministic A-model such that for some $s_0 \in S$, we have $M, s_0 \models \beta_\Delta$. Let $s, s' \in \mathcal{R}(s_0)$ such that $M, s \models D$ and $s \overset{\rho}{\Rightarrow} s'$, where $\rho \in \{x,y\}^*$. Let $m = \sharp_x(\rho)$ and $n = \sharp_y(\rho)$. Then the following assertions hold:*

1. *if $m \geq n$ then $s \overset{(xy)^n x^{m-n}}{\Longrightarrow} s'$*
2. *if $m \leq n$ then $s \overset{(xy)^m y^{n-m}}{\Longrightarrow} s'$*
3. *$m = n$ iff $(M, s' \models D)$.*

Proof: We first prove (1) and (2) by induction on the length of ρ.

The base case, when $\rho = \lambda$ is trivial as m = n = 0, $s \stackrel{\lambda}{\Rightarrow} s$, as required.

For the induction step, let $\rho = \rho' x$ (the proof when $\rho = \rho' y$ is similar). Let s'' be such that $s \stackrel{\rho'}{\Rightarrow} s'' \stackrel{x}{\rightarrow} s'$. Let $m' = \sharp_x(\rho')$. Clearly, $n = \sharp_y(\rho')$ and $m = m' + 1$. There are two cases:

Case 1: $m > n$: Hence $m' \geq n$. By the induction hypothesis (1), $s \stackrel{(xy)^n x^{m'-n}}{\Longrightarrow} s''$ and since we have $s'' \stackrel{x}{\rightarrow} s'$, we get $s \stackrel{(xy)^n x^{m-n}}{\Longrightarrow} s'$ as required.

Case 2: $m \leq n$: Hence $m' < n$. By the induction hypothesis (2), $s \stackrel{(xy)^{m'} y^k}{\Longrightarrow} s''$, where $k = n - m'$. Let $t_0, t_1, \ldots t_{k-1} \in S$ such that

$$s \stackrel{(xy)^{m'}}{\Longrightarrow} t_0 \stackrel{y}{\rightarrow} t_1 \ldots t_{k-1} \stackrel{y}{\rightarrow} s''.$$

Now we have $t_{k-1} \stackrel{y}{\rightarrow} s'' \stackrel{x}{\rightarrow} s'$, hence by Lemma 5.3 for deterministic A-frames, there exists t'_{k-1} such that $t_{k-1} \stackrel{x}{\rightarrow} t'_{k-1} \stackrel{y}{\rightarrow} s'$. By repeating this argument, we can find t'_1 such that $s \stackrel{(xy)^{m'}}{\Longrightarrow} t_0 \stackrel{y}{\rightarrow} t_1 \stackrel{x}{\rightarrow} t'_1 \stackrel{y^{k-1}}{\Longrightarrow} s'$ (refer to Figure 9). Again by Lemma 5.3 for deterministic A-frames, $t_0 \stackrel{\{x,y\}}{\longrightarrow} t'_1$. Thus $s \stackrel{(xy)^m}{\Longrightarrow} t'_1$. Since, $k - 1 = n - m$, we get $s \stackrel{(xy)^m y^{n-m}}{\Longrightarrow} s'$, as required.

Thus (1) and (2) are proved. Now we prove (3).

Suppose $m = n$. By (1) and (2), we get $s \stackrel{(xy)^m}{\Longrightarrow} s'$. We show by induction on m that $M, s' \models D$. The base case when $m = 0$ is trivial, since $m = 0$ and hence $s = s'$ and $M, s \models D$ by assumption of the Lemma. If $m > 0$ then there exists $s'' \in S$ such that $s \stackrel{(xy)^{m-1}}{\Longrightarrow} s'' \stackrel{xy}{\Longrightarrow} s'$. By induction hypothesis, we get $M, s'' \models D$. But $M, s_0 \models \beta_7$ and $s'' \in \mathcal{R}(s_0)$, hence $M, s'' \models [\{x, y\}]D$ and hence $M, s' \models D$, as required.

Suppose $m \neq n$. Then either $m < n$ or $m > n$. Suppose $m < n$. By (ii), we get $s \stackrel{(xy)^m y^k}{\Longrightarrow} s'$, where $k = (n - m) > 0$. Thus we have $t_0, t_1, \ldots t_{k-1} \in S$ such that $s \stackrel{(xy)^m}{\Longrightarrow} t_0 \stackrel{y}{\rightarrow} t_1 \ldots t_{k-1} \stackrel{y}{\rightarrow} s'$. But the proof above tells us that $M, t_0 \models D$. Now, using β_7, we get $M, t_0 \models [y]AD$ and hence $M, t_1 \models AD$. β_8 ensures that $M, t_1 \models [y]AD$. Repeating the argument, we see that $M, s' \models AD$. But then, because of β_6, we get $M, s' \models {\sim}D$. On the other hand, when $m > n$, we use (i) above in a similar fashion to show that $M, s' \models BD$ and thus again appealing to β_6, we get $M, s' \models {\sim}D$. Hence the result. $\square$

Lemma 6.9 *Let $\Delta = (C, R, U, c_r)$ be an instance of RCP such that $\beta_\Delta \in DSAT$. Then Δ has a solution.*

Proof: Let $M, s_0 \models \beta_\Delta$, where $M = (TS, V)$, $TS = (S, \rightarrow)$ is a deterministic A-dts over Σ and $s_0 \in S$.

As before, for constructing a colouring function for Δ, we adapt the following strategy: we first decide the colours on the diagonal in $\mathbf{N} \times \mathbf{N}$ and then inductively fill out larger and larger squares. For each point on the grid, we associate a state in $\mathcal{R}(s_0)$; this is sufficient since the formula $\beta_3 \wedge \beta_4 \wedge \beta_5$ is satisfied at that state and hence the colouring function can be easily "pulled out".

The only complication which arises now is that when we construct the diagonal, we have to ensure that infinitely many points along the diagonal satisfy the proposition RR.

The function $Diag : \mathbf{N} \to \mathcal{R}(s_0)$ is defined inductively. Let $Diag(0) \stackrel{\text{def}}{=} s_0$. Inductively we can assume for $k > 0$, $Diag(k-1) = s \in \mathcal{R}(s_0)$.

By β_2 and β_7, $M, s \models D \wedge \langle\{x,y\}\rangle(D \wedge \Diamond(D \wedge RR))$. Hence, for some ρ such that $|\rho| > 0$, $s \stackrel{\rho}{\Longrightarrow} s'$ and $M, s' \models (D \wedge RR)$. But then by β_2, we find that $\rho \in \{x,y\}^*$.

Now, by Lemma 6.8, we get $s \stackrel{(xy)^m}{\Longrightarrow} s'$, where $m = \sharp_x(\rho) = \sharp_y(\rho)$. Let $t_1, \ldots t_{m-1} \in S$ such that $s \stackrel{xy}{\Longrightarrow} t_1 \ldots t_{m-1} \stackrel{xy}{\Longrightarrow} s'$. Set $t_m = s'$. Clearly, for all $j \in \{1, \ldots m\}$, $M, t_j \models D$. Define $Diag(k-1+j) \stackrel{\text{def}}{=} t_j$, for $j \in \{1, \ldots m\}$.

By induction, $Diag$ is totally defined. Clearly, we have $Diag(i) \stackrel{\{x,y\}}{\longrightarrow} Diag(i+1)$, for all i.

We again construct an infinite sequence of function pairs $\{(\Psi_m, Col_m)\}_{m \geq 0}$ with $\Psi_m : \{0, \ldots, m\} \times \{0, \ldots, m\} \to S$ and $Col_m : \{0, \ldots, m\} \times \{0, \ldots, m\} \to C$ such that the following conditions are satisfied at every stage $m, m \geq 0$:

$(C1)$	$Col_m(0,0) = c_0$	
$(C2)$	$\Psi_m(i,j) \stackrel{x}{\to} \Psi_m(i+1,j)$	$[0 \leq i < m, 0 \leq j \leq m]$
$(C3)$	$\Psi_m(i,j) \stackrel{y}{\to} \Psi_m(i,j+1)$	$[0 \leq i \leq m, 0 \leq j < m]$
$(C4)$	$\Psi_m(i,i) = Diag(i)$	$[0 \leq i \leq m]$
$(C5)$	$Col_m(i+1,j) \in R(Col_m(i,j))$	$[0 \leq i < m, 0 \leq j \leq m]$
$(C6)$	$Col_m(i,j+1) \in U(Col_m(i,j))$	$[0 \leq i \leq m, 0 \leq j < m]$

The construction proceeds exactly as in the proof of Lemma 5.4 and is hence omitted.

Finally define $Col : \mathbf{N} \times \mathbf{N} \to C$ by $Col(i,j) \stackrel{\text{def}}{=} Col_m(i,j)$, where $m = max\{i,j\}$. We now show that $Col(0,j) = c_r$, for infinitely many j; the other conditions on Col are easily seen to be satisfied thanks to the conditions above.

We know that by construction, $M, \Psi_m(m,m) \models RR$ for infinitely many m. Fix any such m. If $m = 0$ then $M, \Psi_m(0,m) \models RR$. Otherwise note that $\Psi_m(m-1,m) \stackrel{x}{\to} \Psi_m(m,m)$ and hence $M, \Psi_m(m-1,m) \models \langle x\rangle RR$. By β_9, $M, \Psi_m(m-1,m) \models RR$. Repeating this argument, we find $M, \Psi_m(0,m) \models RR$. But then by β_{10}, we get $M, \Psi_m(m,m) \models c_r$ as well. Since this is true for infinitely many m, the recurrence condition on Col is satisfied. □

Theorem 6.10 *Suppose $|A| > 1$. Then $DSAT_A$ is Σ_1^1-complete. Hence $DVAL_A$ is a Π_1^1-complete set and not axiomatizable.*

Proof: By the earlier Lemma 6.7 and Lemma 6.9, any instance Δ of RCP has a solution iff the formula $\beta_\Delta \in DSAT_A$. Since RCP is Σ_1^1-complete [Parikh], so is membership in $DSAT_A$. □

This negative result is extended to trace languages in Section 8.

7 Finite DTS's

An important and interesting subclass of dts's is that of finite dts's. Recall that the dts $TS = (S, \Sigma, \rightarrow)$ is said to be finite if and only if *both* S and $\rightarrow$ are finite sets. Clearly if $TS = (S, \Sigma, \rightarrow)$ is a dts over A, where $A \in \wp_{fin}(\Sigma)$ then $\rightarrow$ is finite whenever S is finite. In general, we could have S finite and $\rightarrow$ infinite. One result we will show here is that our logic cannot distinguish between these two situations even in the presence of determinacy. As a result, it suffices to deal with just the strong notion of finiteness, where both S and $\rightarrow$ are finite.

We say that a formula α has a *finite* model (that is, a model based on a finite frame) iff there exists a finite model $M = ((S, \rightarrow), V)$ and $s \in S$ such that $M, s \models \alpha$. Let $FSAT$ denote the set of all formulas which have finite models and let $FVAL$ denote the set of formulas that are valid over the class of finite models. Then $FDSAT$ and $FDVAL$ will denote the relevant sets of formulas with reference to finite deterministic models. The sets $FSAT_A$, $FVAL_A$, $FDSAT_A$ and $FDVAL_A$, where $A \in \wp_{fin}(\Sigma)$, are defined in the obvious way.

Firstly, we review all our earlier results in the context of finite models. The system ND is easily seen to be a sound and complete axiomatization of $FVAL$; finiteness of models does not disturb soundness, and the completeness proof in Section 2 (Theorem 2.7) does produce a finite model for any ND-consistent formula. Similar remarks apply for ND_A and $FVAL_A$.

Turning now to the results of Section 3, it is clear that the proof of Theorem 3.4 cannot work if we insist on finite models based on event structures: since event structures are poset-based, a formula such as $\Box\langle a\rangle True$ will necessarily require a model based on an infinite event structure. However, the problem is open in the case of elementary net systems. We do not know whether for every formula in $FSAT$, there exists a model $M = (TS, V)$ such that $TS = TS_{\mathcal{N}}$ for some finite labelled elementary net system $\mathcal{N}$.

Before turning to $FDSAT$, we show that our logic cannot distinguish between finite dts's and finite state dts's, deterministic or otherwise:

Proposition 7.1 *Let $M = ((S, \rightarrow), V)$ be a model, S a finite set, $s_0 \in S$ and $M, s_0 \models \alpha$. Then*

1. *$\alpha \in FSAT$.*
2. *Suppose M is a deterministic model. Then $\alpha \in FDSAT$.*

Proof: We prove only part (2); the other proof follows. Assume M, s_0, α to be given. First fix an *injective* function $f : S \times S \rightarrow (\Sigma - Voc(\alpha))$. The existence of f is assured since Σ is infinite whereas both $S \times S$ and $Voc(\alpha)$ are finite. Define $TS' \stackrel{\text{def}}{=} (S, \rightarrow')$ where

$$\rightarrow' \stackrel{\text{def}}{=} \{(s, u, s') \mid s \stackrel{u}{\rightarrow} s' \text{ and } u \subseteq Voc(\alpha)\}$$
$$\cup\{(s, \{f(s, s')\}, s') \mid s \stackrel{a}{\rightarrow} s' \text{ and } a \notin Voc(\alpha)\}$$

It is easy to verify that TS' is a dts. Determinacy of TS' follows from that of TS and the injectiveness of f. Further TS' is *finite* since S was assumed to be finite and $\rightarrow'$ is finite by construction. From the definition of $\rightarrow'$, we can make the following crucial remark about TS':

$$\forall s, s' \in S : \; s' \in \mathcal{R}_{TS}(s) \text{ iff } s' \in \mathcal{R}_{TS'}(s).$$

Consider $M' \stackrel{\text{def}}{=} (TS', V)$.

Claim : $\forall s \in S : \forall \beta \in CL(\alpha) : M, s \models \beta$ iff $M', s \models \beta$.

The proof of the claim proceeds by an easy induction on the structure of β and is omitted here. Since $M, s_0 \models \alpha$, by the claim above, we have $M', s_0 \models \alpha$ as well. Hence $\alpha \in FDSAT$. □

The decision procedure given in Section 2 also shows that the membership problem for $FSAT$ is decidable in nondeterministic exponential time. In the case of $DSAT$ we showed undecidability in Section 5. However, we do not know whether the membership problem for $FDSAT$ is decidable or not. We do know, thanks to Proposition 7.1, that $FDSAT = \bigcup_{A \in \wp_{fin}(\Sigma)} FDSAT_A$. Moreover, we can also easily deduce that $FDSAT$ is a recursively enumerable set. Hence $FDVAL$ is at worst co-r.e. But it might well be the case that $FDSAT$ is not recursive, in which case $FDVAL$ would not be r.e. and hence not axiomatizable.

On the other hand, when $A \in \wp_{fin}(\Sigma)$, $|A| > 1$, we can show an undecidability result for $FDSAT_A$. We show this with yet another variant of the colouring problem called the *Finite Colouring Problem* (FCP for short).

An *instance* of FCP is a triple $\Delta = (C, R, U, c_f)$ where $C = \{c_0, c_1, \ldots, c_k\}$ is a finite non-empty set of colours such that $c_f \in C$ and $R, U : C \to \wp(C)$ are the "right" and "up" functions as before. A *solution* to Δ is a pair $(Col, (K, L))$, where $K, L \in \mathbf{N}$ and $Col : \{0, \ldots, K\} \times \{0, \ldots, L\} \to C$ is a colouring function which satisfies:

1. $Col(0,0) = c_0$.
2. $Col(i+1, j) \in R(Col(i,j)),\ 0 \le i < K,\ 0 \le j \le L$.
3. $Col(i, j+1) \in U(Col(i,j)),\ 0 \le i \le K,\ 0 \le j < L$.
4. $Col(K, L) = c_f$.

Proposition 7.2 *FCP is undecidable.*

Proof: (Sketch) We can reduce to FCP the halting problem of Turing machines started on a blank tape with the head on the leftmost cell. Each such TM can be coded as an instance Δ_{TM} of FCP. The coding scheme closely follows the one given in [LP]. We can then show that TM halts if and only if Δ_{TM} has a solution. □

We now reduce each instance of FCP to a membership problem for $FDSAT_A$. In other words, we shall encode each instance Δ of FCP into a formula γ_Δ such that Δ has a solution iff $\gamma_\Delta \in FDSAT_A$. It is assumed that $|A| > 1$. We will ensure that γ is an A-formula. Without loss of generality, let $x, y \in A$ and as before, we reserve x and y respectively for R and U. As usual, we let $C \subseteq P$. In addition, we use two special propositions UM and RM respectively for "up-margin" and "right-margin".

Definition 7.3 *Let* $\Delta = (C, R, U, c_f)$ *be an instance of FCP, where* $C = \{c_0, c_1, \ldots, c_k\}$ *and* $c_f \in C$. *Then,* $\gamma_\Delta \stackrel{\text{def}}{=} \bigwedge_{i=1}^{6} \gamma_i$, *where*

1. $\gamma_1 \stackrel{\text{def}}{=} c_0 \wedge \Diamond(c_f \wedge UM \wedge RM)$.
2. $\gamma_2 \stackrel{\text{def}}{=} \Box(\langle\{x,y\}\rangle True \equiv (\sim UM \wedge \sim RM)) \wedge \Box(\bigwedge_{d \in (A - \{x,y\})} [d]False)$.
3. $\gamma_3 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \equiv \bigwedge_{j \neq i} \sim c_j)$.
4. $\gamma_4 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \supset [x] \bigvee_{c \in R(c_i)} c)$.
5. $\gamma_5 \stackrel{\text{def}}{=} \Box \bigwedge_{i=0}^{k} (c_i \supset [y] \bigvee_{c \in U(c_i)} c)$.
6. $\gamma_6 \stackrel{\text{def}}{=} \Box((UM \supset [x]UM \wedge [y]False) \wedge (RM \supset [y]RM \wedge [x]False))$.

The first clause, apart from capturing the origin constraint, also specifies a termination condition. The second clause forces the creation of a grid as in the earlier reductions, but this time only upto an upper margin (UM) and a right margin (RM). The next three clauses are familiar. The last clause ensures that the propositions UM and RM acquire their intended meaning.

Lemma 7.4 *Let* $\Delta = (C, R, U, c_f)$ *be an instance of FCP. If* Δ *has a solution, then* $\gamma_\Delta \in FDSAT_A$.

Proof: Let $(Col, (K, L))$ be a solution to FCP. Define now $TS = (S, \rightarrow)$ as in the proof of Lemma 5.2, but now for $S = \{0, \ldots, K\} \times \{0, \ldots, L\}$. Next define $V : S \rightarrow \wp(P)$ to be a map which satisfies, for all $i, j \in \{0, \ldots, K\} \times \{0, \ldots, L\}$:

- $V(i,j) \subseteq Col(i,j) \cup \{UM, RM\}$.
- $RM \in V(i,j)$ iff $i = K$.
- $UM \in V(i,j)$ iff $j = L$

Clearly, V is a well-defined map. Let $M = (TS, V)$. Then it is easy to show that $M, (0,0) \models \gamma_\Delta$. Hence $\gamma_\Delta \in FDSAT_A$. $\Box$

Lemma 7.5 *Let* $M = (TS, V)$ *be a model where* $TS = (S, \rightarrow)$ *is a finite deterministic dts over* A *and* $s_0 \in S$ *such that* $M, s_0 \models \gamma_\Delta$ *where* Δ *is an instance of FCP. Let* $s, s' \in \mathcal{R}(s_0)$. *Then the following statements are equivalent:*

1. $s \xrightarrow{\{x,y\}} s'$.

2. $\exists s_x \in S : s \xrightarrow{x} s_x \xrightarrow{y} s'$.

3. $\exists s_y \in S : s \xrightarrow{y} s_y \xrightarrow{x} s'$.

Proof: (1) implies (2) and (3) since TS is a dts. To show that (2) implies (1), assume $s \xrightarrow{x} s_x \xrightarrow{y} s_y$. Then $M, s_x \models \langle y \rangle True$. Now, because of γ_6, $M, s_x \models (UM \supset [y]False)$. Therefore $M, s_x \models \sim UM$. Further, $M, s \models (UM \supset [x]UM)$ and so, $M, s \models \sim UM$. From the fact that $s \xrightarrow{x} s_x$, we get $M, s \models \langle x \rangle True$ and thanks to γ_6, we have $M, s \models \sim RM$. Thus, $M, s \models (\sim UM \wedge \sim RM)$. Now, by γ_2, we get $M, s \models \langle \{x,y\} \rangle True$. Therefore, for some $s'' \in S$, we have $s \xrightarrow{\{x,y\}} s''$. Hence there exists s'_x such that $s \xrightarrow{x} s'_x \xrightarrow{y} s''$. By determinacy of TS, we get $s_x = s'_x$ and hence $s' = s''$.

By a symmetric argument we can show that (3) implies (1) as well. $\square$

Lemma 7.6 *Let Δ be an instance of FCP and $M = ((S, \rightarrow), V)$ be a finite deterministic A-model such that for some $s_0 \in S$, we have $M, s_0 \models \gamma_\Delta$. Let $s, s' \in \mathcal{R}(s_0)$ such that $s \xRightarrow{\rho} s'$, where $\rho \in \{x,y\}^*$. Let $m = \sharp_x(\rho)$ and $n = \sharp_y(\rho)$. Then the following assertions hold:*

1. *if $m \geq n$ then $s \xRightarrow{(xy)^n x^{m-n}} s'$*

2. *if $m \leq n$ then $s \xRightarrow{(xy)^m y^{n-m}} s'$*

Proof: Identical to the proof of (1) and (2) of Lemma 6.8, except that instead of appealing to Lemma 5.3, we refer to Lemma 7.5 above. $\square$

Lemma 7.7 *Let $\Delta = (C, R, U, c_f)$ be an instance of FCP such that $\gamma_\Delta \in FDSAT_A$. Then Δ has a solution.*

Proof: Let $M, s_0 \models \gamma_\Delta$, where $M = (TS, V), TS = (S, \rightarrow)$ is a finite deterministic dts over A and $s_0 \in S$. Since $M, s_0 \models \Diamond(c_f \wedge UM \wedge RM)$, there exists $s_1 \in S$ and ρ such that $s_0 \xRightarrow{\rho} s_1$. γ_2 ensures that $\rho \in \{x,y\}^*$. Let $m = \sharp_x(\rho)$ and let $n = \sharp_y(\rho)$. We have three cases to consider:

Case 1 ($m = n$): By Lemma 7.6 above $s_0 \xRightarrow{(xy)^m} s_1$. Let $t_1, \ldots, t_{m-1} \in S$ such that $s_0 = t_0 \xRightarrow{xy} t_1 \ldots t_{m-1} \xRightarrow{xy} t_m = s_1$. Define $Diag : \{0, \ldots, m\} \rightarrow S$ by $Diag(k) \stackrel{\text{def}}{=} t_k$. Following the proof of Lemma 5.4, we can construct a function pair (Ψ_m, Col_m) with $\Psi_m : \{0, \ldots, m\} \times \{0, \ldots, m\} \rightarrow S$ and $Col_m : \{0, \ldots, m\} \times \{0, \ldots, m\} \rightarrow C$ such that Col_m is a solution to Δ.

Case 2 ($m < n$): By Lemma 7.6 $s_0 \xRightarrow{(xy)^m y^{n-m}} s_1$. Let $s' \in S$ such that $s_0 \xRightarrow{(xy)^m} s' \xRightarrow{y^{n-m}} s_1$. Again we follow Lemma 5.4. Construct (Ψ_m, Col_m) as in the proof for Case 1, with $\Psi_m(0,0) = s_0$ and $\Psi_m(m,m) = s'$. (Note that we no longer maintain C2 of Lemma 5.4.) Let $k = n - m$. Let $t_1, \ldots, t_{k-1} \in S$ such that $s' = t_0 \xrightarrow{y} t_1 \ldots t_{k-1} \xrightarrow{y} t_k = s_1$.

Now, we define for $l \in \{1, \ldots, k\}, \Psi_{m+l} : \{0, \ldots, m\} \times \{0, \ldots, m+l\} \to S$ and $Col_{m+l} : \{0, \ldots, m\} \times \{0, \ldots, m+l\} \to C$. Firstly set

$$\Psi_{m+l}(i,j) \stackrel{\text{def}}{=} \Psi_{m+l-1}(i,j)$$

and

$$Col_{m+l}(i,j) \stackrel{\text{def}}{=} Col_{m+l-1}(i,j), \quad 0 \leq i \leq l-1, \quad 0 \leq j \leq l-1.$$

Next set $\Psi_{m+l}(m, m+l) \stackrel{\text{def}}{=} t_l$. Now we have

$$\Psi_{m+l}(m-1, m+l-1) \xrightarrow{x} \Psi_{m+l}(m, m+l-1) \xrightarrow{y} \Psi_{m+l}(m, m+l).$$

Hence there exists $s'' \in S$ such that

$$\Psi_{m+l}(m-1, m+l-1) \xrightarrow{y} s'' \xrightarrow{x} \Psi_{m+l}(m, m+l).$$

Set $\Psi_{m+l}(m-1, m+l) \stackrel{\text{def}}{=} s''$. Repeating this argument, we define $\Psi_{m+l}(j, m+l)$ for all $j, 0 \leq j \leq m$. $Col_{m+l}(j, m+l)$ can be suitably defined for $0 \leq j \leq m$.

It can be easily checked that Col_{m+k}, that is Col_n is a solution to Δ.

Case 3 ($m > n$): Similar to the proof for Case 2. This time we follow Lemma 5.4 but do not maintain condition C3. □

Theorem 7.8 *Let $|A| > 1$. Then the membership problem for $FDSAT_A$ is undecidable. Consequently, $FDVAL_A$ is not axiomatizable.*

Proof: The undecidability follows from Proposition 7.2, and Lemmas 7.4 and 7.7. It is easy to see that $FDSAT_A$ is r.e. and hence $FDVAL_A$ is not r.e. and therefore not axiomatizable. □

8 Traces and trace transition systems

In this section, we show that our proof methods yield results for transition systems based on the theory of *trace languages* [Maz]. Specifically, we shall show that the satisfiability problem for our logic becomes undecidable when it is interpreted over models based on *trace* transition systems. In fact, the result holds for a much weaker logical language – the eventuality operator of temporal logic with an action-indexed modality suffices to establish undecidability. We can extend the result to subclasses as in the previous two sections.

As we noticed, our proofs of undecidability rely on a weaker property than determinacy, specified in Lemma 5.3. In particular, the partial commutativity of actions gives rise to the same phenomenon. In concurrency theory, this arises in the context of Mazurkiewicz's trace languages. Here we present only the bare essentials of this theory. For background and more details, refer to [Maz].

A *concurrency alphabet* over Σ is a pair (Σ, I), where $I \subseteq \Sigma \times \Sigma$ is an irreflexive and symmetric *independence relation*. Our results will require the concurrency alphabet to be *nontrivial*, that is, I has to be a nonempty independence relation. Note that this forces $|\Sigma| > 1$.

The independence relation I induces a natural equivalence relation over Σ^* which is in fact a congruence with respect to concatenation. This congruence is the one generated by equations of the form $ab = ba$ for each $(a, b) \in I$. Stated differently, we first define $\doteq_I \subseteq \Sigma^* \times \Sigma^*$ as: $\rho \doteq_I \rho'$ iff $\exists \rho_1, \rho_2 \in \Sigma^*$ and $(a, b) \in I$ such that $\rho = \rho_1 ab \rho_2$ and $\rho' = \rho_1 ba \rho_2$. Then $=_I$, defined to be $(\doteq_I)^*$ is the congruence we want. $\Sigma^*/{=_I}$ is called the *partially commutative trace monoid* over (Σ, I) (with $[\rho]_I.[\rho']_I = [\rho\rho']_I$ being the monoidal operation). A *trace language* over (Σ, I) is simply a subset of $\Sigma^*/{=_I}$.

Thus the idea is that if $a\ I\ b$, then whenever a and b occur adjacent to each other in a sequential description of a run of the system (modelled by the trace language), a and b have in fact occurred with no order over their occurrences. Hence a sequence of the form $\rho_1 ab \rho_2$ represents the *same* stretch of behaviour as a sequence of the form $\rho_1 ba \rho_2$.

A number of closely related proposals have been made in the literature to carry over these ideas to transition systems [Bed, Shi, WN]. We define a class of transition systems for which the only constraint is the commuting of sequences of concurrent actions. This suffices for our purpose, and our negative results will carry over to the transition systems defined in the above papers.

Definition 8.1 *A* **trace transition system (tts)** *over the concurrency alphabet* (Σ, I) *is a (countable) labelled transition system* $TS = (S, \Sigma, \rightarrow)$ *such that for every* $(a, b) \in I$*, for every* $s_0, s_1, s_2 \in S$*, if* $s_0 \xrightarrow{a} s_1 \xrightarrow{b} s_2$ *then there exists* s_1' *such that* $s_0 \xrightarrow{b} s_1' \xrightarrow{a} s_2$*.*

Instead of Step-TL, we now work with the simpler language Action-TL, which has the $\Diamond$ modality as usual and the action modality $\langle a \rangle$ for every $a \in \Sigma$. Let P be a countable set of propositions. The formulas of this language are:

- Every member of P is a formula.
- If α and β are formulas then so are $\sim\alpha, \alpha \vee \beta, \Diamond\alpha$ and $\langle a \rangle \alpha$, for $a \in \Sigma$.

The semantics is defined as before. For a tts-based model $M = ((S, \rightarrow), V)$ and $s \in S$, we have:

$M, s \models \langle a \rangle \alpha$ iff there exists s' such that $s \xrightarrow{a} s'$ and $M, s \models \alpha$.

Clearly, Action-TL is a weaker language than Step-TL; in fact, it corresponds to the formulas of Step-TL where steps are restricted to be of size 1.

Definition 8.2 *Let* (Σ, I) *be a concurrency alphabet.*

- α *is said to be* I-satisfiable *iff there exists a model* $M = (TS, V)$*, where* $TS = (S, \rightarrow)$ *is a tts over* (Σ, I) *and* $s_0 \in S$ *such that* $M, s_0 \models \alpha$*.*
- $TSAT_I$ *is the set of all I-satisfiable formulas.*

- *We write* $\models_I \alpha$ *if* α *is valid over all models over* (Σ, I).

Given a nonempty independence relation I, we show undecidability of I-satisfiability, again by reducing CP to it. Let $\Delta = (C, R, U)$ be an instance of CP. We need to reserve two actions from Σ for R and U. We choose x and y, where $(x, y) \in I$. Below, whenever appropriate, we follow the notations and conventions used in proving Theorem 5.5. As before, α_Δ is the conjunction of five formulas, except that we modify α_2 to be $\Box\langle x\rangle\langle y\rangle True$.

Lemma 8.3 *Let* $\Delta = (C, R, U)$ *be an instance of CP. If* Δ *has a solution, then* $\alpha_\Delta \in TSAT_I$.

Proof: From the dts constructed in the proof of Lemma 5.2, one can clearly extract a tts over (Σ, I) by forgetting the $\{x, y\}$ transitions. Hence $\alpha_\Delta \in TSAT_I$. □

Lemma 8.4 *Let* $\Delta = (C, R, U)$ *be an instance of CP such that* $\alpha_\Delta \in TSAT_I$. *Then* Δ *has a solution.*

Proof: Let $M, s_0 \models \alpha_\Delta$, where $M = (TS, V)$, $TS = (S, \rightarrow)$ is a tts over (Σ, I) and $s_0 \in S$. By definition, $\mathcal{R}(s_0)$ is countable. Fix an enumeration of $\mathcal{R}(s_0)$.

We proceed exactly as in the proof of Lemma 5.4. Instead of Lemma 5.3, we appeal directly to the definition of a trace transition system. The few modifications required are as follows:

1. In Step 2, when choosing $\Psi_{m+1}(m+1, m+1)$, set it equal to s, where s is the state with the least index (in the enumeration of $\mathcal{R}(s_0)$) with the property that $\Psi_m(m, m) \xrightarrow{\{x,y\}} s$.
2. In Step 3, when choosing $\Psi_{m+1}(m+1, j)$, for $0 \le j \le m$, appeal to Lemma 8.4 instead of Lemma 5.3 and set it equal to s_y, where s_y is the state with the least index (in the enumeration of $\mathcal{R}(s_0)$) with the property that $\Psi_{m+1}(m, j) \xrightarrow{y} s_y \xrightarrow{x} \Psi_{m+1}(m+1, j+1)$. A similar modification is done for the choice of s_x in Step 4.

The required result now follows easily. □

Theorem 8.5 *Let* (Σ, I) *be a nontrivial concurrency alphabet. I-satisfiability is undecidable.*

What about an axiomatization? The following is a sound axiom system. All the axioms are derived from the earlier axiomatization, but now restricted to the language Action-TL. The only novelty is in the axiom (A_{ab}) which represents the commuting condition for a and b.

Axiom System NT_I

Axiom schemes

$(A0)$ All the substitutional instances of the tautologies of PC

$(A1)$ $\Box(\alpha \supset \beta) \supset (\Box\alpha \supset \Box\beta)$

$(A2)$ $\Box\alpha \supset \alpha \wedge [a]\alpha \wedge \Box\Box\alpha$

$(A3)$ $[a](\alpha \supset \beta) \supset ([a]\alpha \supset [a]\beta)$

(A_{ab}) $\langle a\rangle\langle b\rangle\alpha \supset \langle b\rangle\langle a\rangle\alpha,$ for $a \; I \; b$

Inference rules

(MP) $\dfrac{\alpha, \quad \alpha \supset \beta}{\beta}$ $\quad (TG)$ $\dfrac{\alpha}{\Box\alpha}$

If I is a *finite* relation, we can show that I-validity is completely axiomatized by NT_I.

Theorem 8.6 (Completeness) *Given a concurrency alphabet* (Σ, I) *where* I *is a finite independence relation on* Σ, *if* $\models_I \alpha$ *then* $\vdash_{NT_I} \alpha$.

Proof: The proof follows along the lines of that of Theorem 4.9. When satisfying live future requirements, we pick an action d which is outside the vocabulary of α and which, in addition, does not commute with any other action in Σ. Since I is finite and Σ is countable, this is always possible. $\Box$

Consider now the case that the alphabet Σ is finite. We get the stronger undecidability result of Section 6. Since the techniques involved are very similar to the ones used earlier, we will give only an informal sketch of the proof.

Given an instance Δ of RCP, we define the formula β_Δ as before except that β_2 is defined to be $\Box(\langle x\rangle\langle y\rangle True \wedge \bigwedge_{a \notin \{x,y\}} [a]False)$. It is easy to show that β_Δ is I-satisfiable, where (without loss of generality) $(x, y) \in I$. To see this, we only need to extract from the dts constructed in the proof of Lemma 6.7 a tts over $(\{x, y\}, \{(x, y), (y, x)\})$.

On the other hand, given a model for the formula β_Δ, to construct a solution the instance Δ of RCP, one has to simply go through the steps in the proof of Lemma 6.9, making the necessary modifications as suggested in the proof of Lemma 8.4, using the fact that $\mathcal{R}(s_0)$ is enumerable (where s_0 is the state at which the formula β_Δ is satisfied in the given model). Indeed, the proof of Lemma 6.9 follows the given lines only so that it applies for tts's as well.

Theorem 8.7 *Let* (Σ, I) *be a nontrivial concurrency alphabet over finite* Σ. *Then* I*-satisfiability is* Σ_1^1*-complete. Hence* I*-validity is* Π_1^1*-complete and not axiomatizable.*

Similarly, we can consider *finite* trace transition systems. The corresponding satisfiability problem is undecidable and hence validity is not axiomatizable.

Theorem 8.8 *Let* (Σ, I) *be a nontrivial concurrency alphabet. Then* I*-satisfibiality over finite tts's is undecidable and validity is not axiomatizable.*

An analogue of Theorem 8.7 is already available in [Har84], but in the context of the *global consequence* problem of PDL. The corresponding notion of transition systems would be those which satisfied

$$s \xrightarrow{ab} s' \text{ and } s \xrightarrow{ba} s'' \text{ implies } s' = s''.$$

This would be the case for *deterministic* tts's.

Our result for Action-TL shows that even with nondeterminism allowed, the commuting condition of trace transition systems makes even a very weak logic highly expressive. On the other hand, Step-TL – and even the stronger logics considered in the next section – remain decidable over nondeterministic distributed transition systems, where concurrency is explicitly presented rather than being semantically inferred.

9 Extensions

In this section we look at some different logical languages for the frames we have been considering. The two extensions we consider are to allow *program* operators in place of the temporal $\Diamond$, and to strengthen the step modality to refer to intermediate states in the cube.

9.1 Regular programs over concurrent steps

The notion of a step can be used to obtain a straightforward generalization of Propositional Dynamic Logic (PDL) [Har84]. The resulting language, which we shall call Step-PDL, is closely related to the language used so far. Most of the results we have proved so far go through for Step-PDL with suitable modifications.

First we can define the class of *programs* Π_Σ:

- Every member of $\wp_{fin}(\Sigma)$ is a program.
- If π and π' are programs, then so are $\pi + \pi', \pi; \pi'$ and π^*.

Now the language of Step-PDL consists of the set of formulas built from Π_Σ and P, a countably infinite set of atomic propositions, by closing under negation, disjunction and the modality $\langle \pi \rangle \alpha$, for $\pi \in \Pi_\Sigma$. PDL is usually defined with a test operator, but we do not include it here for the sake of simplicity.

As Kripke frames for Step-PDL, we will once again use dts's. To do so, we first need to extend the step transition relation of a dts to a program transition relation.

Let $TS = (S, \Sigma, \rightarrow)$ be a dts. Then $\Rightarrow_{TS} \subseteq S \times \Pi_\Sigma \times S$ is defined inductively as follows (we drop the TS subscript for convenience):

- $s \stackrel{u}{\Rightarrow} s'$ iff $s \stackrel{u}{\rightarrow} s'$.
- $s \xRightarrow{\pi+\pi'} s'$ iff $s \xRightarrow{\pi} s'$ or $s \xRightarrow{\pi'} s'$.
- $s \xRightarrow{\pi;\pi'} s'$ iff $\exists s'' \in S : s \xRightarrow{\pi} s'' \xRightarrow{\pi'} s'$.
- $s \xRightarrow{\pi^*} s'$ iff $\exists k \geq 0 : s \xRightarrow{\pi^k} s'$, where $\pi^0 \stackrel{\text{def}}{=} \emptyset$ and $\pi^{k+1} \stackrel{\text{def}}{=} \pi; \pi^k$, for $k \geq 0$.

The notions of frame and model are as before. The notion $M, s \models \alpha$, for $s \in S$ is defined inductively, the new case being:

$$M, s \models \langle \pi \rangle \alpha \text{ iff } \exists s' \in S : s \overset{\pi}{\Rightarrow}_{TS} s' \text{ and } M, s \models \alpha.$$

Satisfiability and validity are defined as before. One crucial observation here is that for Step-PDL it makes no difference whether the frames are dts's over Σ or dts's over some finite subset of Σ. (For the negative results, of course, we need $|A| > 1$).

A complete axiomatization of the set of valid formulas of Step-PDL is obtained by adding the empty step axiom $\alpha \equiv \langle \emptyset \rangle \alpha$ and the (*Step*) inference rule to the well-known Segerberg axioms for PDL [KP, Har84]. As a consequence, satisfiability in Step-PDL is decidable in nondeterministic exponential time.

It can be easily checked that the completeness results for elementary net systems and elementary event structures presented in Section 3 go through for Step-PDL. As for the negative results, we do not get an axiomatization of the set of deterministically valid formulas as deterministic satisfiability for Step-PDL formulas is Σ_1^1-complete. Hence deterministic validity is not axiomatizable. (In the coding of RCP, we uniformly replace $\Box$ by $[(x+y)^*]$ and $\Diamond$ by $\langle (x+y)^* \rangle$.)

The strong negative result goes through for trace transition systems as well. Further in the case of finite deterministic dts's and finite trace transition systems, once again the negative result obtains, using the same transformation in the formulas used for coding earlier.

We conclude by noting that instead of generalizing the atomic programs of PDL to concurrent steps, we could also generalize them to finite *multisets* of actions. We could in fact consider finite *pomsets* over Σ [Pra86] to be our atomic programs. Correspondingly, we would have to index the modality by finite multisets or by finite pomsets. In each case, there is a corresponding (and notationally more complicated) version of the inference rule (*Step*) which leads to completeness, and as a by-product, to decidability. Naturally, the negative results we have obtained will also go through.

9.2 Referring to intermediate states

One drawback of the logical languages we have looked at so far is that we have been unable to axiomatize our models with a finite set of axiom schemes and inference rules. By considering a more expressive modality for the u-cube, however, we can overcome this difficulty. We shall merely give a sketch of the main ideas; the details can be worked out.

Given a set of atomic propositions P, the formulas of the language Cube-TL are inductively specified as:

- Every member of P is a formula.
- If α and β are formulas then so are $\sim\alpha, \alpha \vee \beta, \Diamond\alpha$.
- Let $u \in \wp_{fin}(\Sigma)$. If $\alpha_\emptyset, \ldots, \alpha_v, \ldots, \alpha_u$ are formulas ($v \subseteq u$), then $\langle u \rangle < \alpha_\emptyset, \ldots, \alpha_u >$ is a formula.

The last clause defines a formula $\langle u \rangle \Psi$, where Ψ can be viewed as a function from $\wp(u)$ to formulas, where $\Psi(v) = \alpha_v$. The formula states that there exists a u-cube with the states in the cube satisfying the corresponding formulas from Ψ.

Now given a model $M = ((S, \rightarrow), V)$ and $s \in S$,

$$M, s \models \langle u \rangle \Psi \text{ iff } \exists f \in \mathcal{F}[u, S] : f(\emptyset) \xrightarrow{u} f(u),\ f(\emptyset) = s \text{ and } M, f(v) \models \Psi(v) \text{ for } v \subseteq u.$$

That is, the formula $\langle u \rangle \Psi$ forces the existence of a u-cube with intermediate states satisfying the formulas from Ψ.

Observe that Ψ is at least exponential in the size of u. Our earlier modality in the language Step-TL, $\langle u \rangle \alpha$, is *defined* to be $\langle u \rangle \Gamma$, where $\Gamma(u) = \alpha$ and $\Gamma(v) = True$, for $v \subset u$.

Given $v \subseteq v' \subseteq u$ and a function Ψ from $\wp(u)$ to formulas, define its restriction $\Psi_{v..v'}$ to be a function assigning formulas to $\wp(v' - v)$: $\Psi_{v..v'}(u_1) \stackrel{\text{def}}{=} \Psi(v \cup u_1)$, for $u_1 \subseteq v' - v$.

The step axioms and inference rule are:

$(A4a)$ $\quad \alpha \supset \langle \emptyset \rangle < \alpha >$
$(A4b)$ $\quad \langle u \rangle \Psi \supset \Psi(\emptyset)$

$$(Step) \quad \frac{\sim\langle v' - v \rangle \Psi_{v..v'}, \quad for\ some\ v \subseteq v' \subseteq u}{\sim\langle u \rangle \Psi}$$

With these axioms and rule, completeness and decidability can be proved along the lines of Section 2. Since Cube-TL is more expressive than Step-TL, all the negative results for that language will go through.

10 Discussion

In this paper we have studied logics whose models are distributed transition systems of a certain kind. The central notion underlying these transition systems is that of a concurrent step. The properties that are demanded of a step capture the intuition that the actions named in the step occur causally independent of each other. The paper is then essentially a logical study of this basic notion concerning distributed systems.

The main results of the paper are summarized in the table, where we have fixed a countable alphabet Σ and a finite subset A of Σ.

In addition, we have shown that the logical system ND is a complete axiomatization of validity over the class of labelled prime event structures and hence over the class of labelled elementary net systems as well.

Our positive results show that the step notion lends itself to a logical treatment with the help of fairly standard techniques. In fact, as the ideas sketched in Section 9 show, the logic Step-TL itself can be viewed as a smooth extension of PDL in the presence of steps.

On the other hand, our negative results show that from a logical standpoint, determinacy combined with a non-interleaved notion of a transition is very expressive. The results of Section 8 provide additional insight: since the negative results carry over for trace transition systems, we

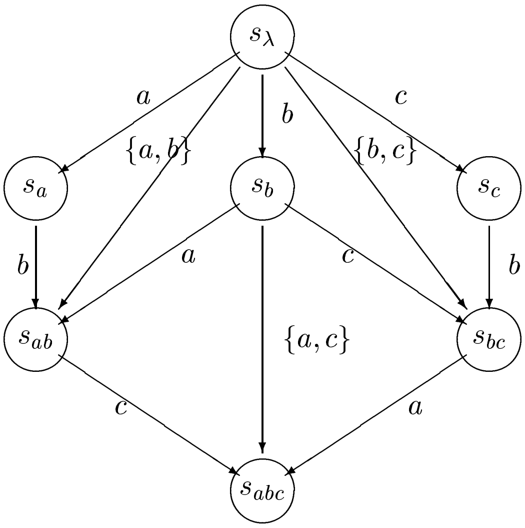
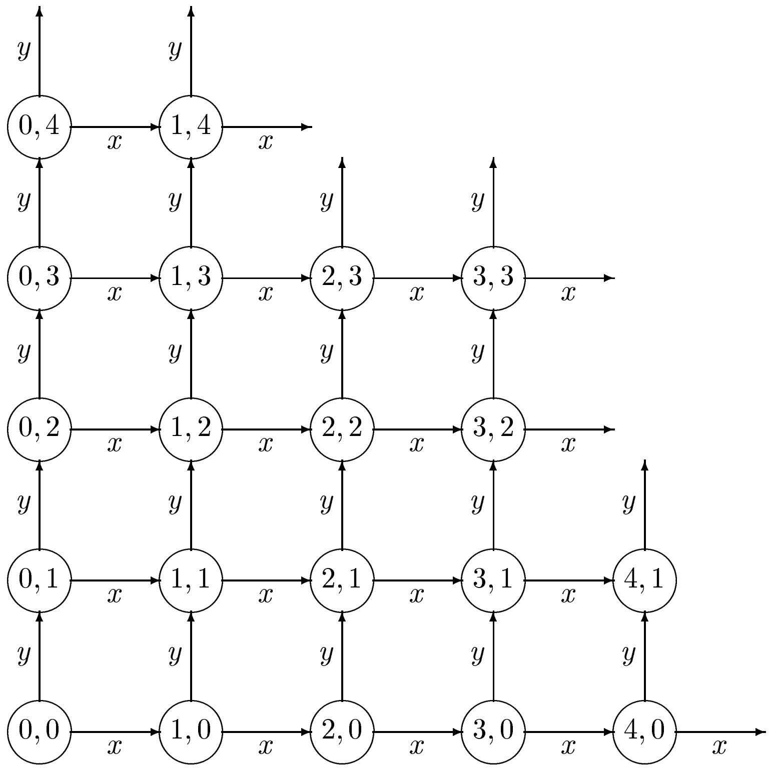


Figure 7: A dts which cannot be generated from an event structure

frames	*all models*	*finite models*
dts's	axiomatizable decidable	axiomatizable decidable
det dts's	axiomatizable undecidable	? at most r.e.
trace ts's over (Σ, I)	axiomatizable for finite I undecidable	? at most r.e.
A-dts's	axiomatizable decidable	axiomatizable decidable
det A-dts's	not axiomatizable highly undecidable	not axiomatizable undecidable
trace ts's over (A, I)	not axiomatizable highly undecidable	not axiomatizable undecidable

Table 1: Step-TL: axiomatizability and satisfiability



$\Box\langle\{x, y\}\rangle True$

Figure 8: $\mathbf{N} \times \mathbf{N}$

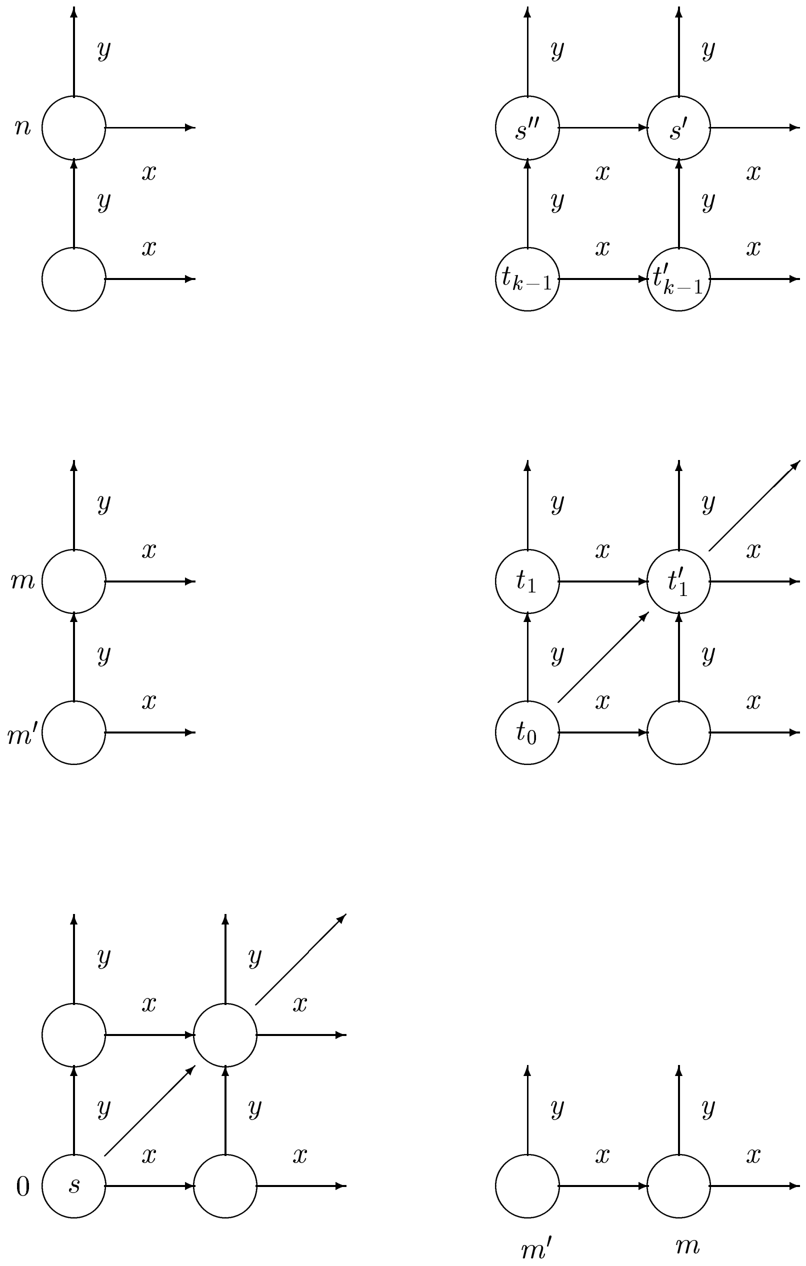


Figure 9: Case 2 of Lemma 6.8

can see that it is not just determinacy together with "non-interleaved" transitions that generates such expressive power; even the kind of partial commutativity of actions that is often associated with independent actions leads to undecidability.

Turning now to related work, Valiev [Val] presents a strong negative result for a variant of PDL. In this variant one has, in addition to the usual program constructs of PDL, also the shuffle and the iterated shuffle operators. The techniques used here are very different from Valiev's work.

Penzcek [Pen] has also reported a number of negative results for a logic interpreted over deterministic asynchronous transition systems. The logical language uses past operators. The results of Section 8 show that the negative results need neither determinacy nor the past time modalities.

As for other logics based on labelled transition systems, two well-known instances are the Hennessy-Milner logics [HM] and the Modal μ-Calculus [Sti]. We have not yet "operationally" characterized (in the Hennessy-Milner style) the equivalence notion induced by our logic. It is also not clear at this stage whether the Modal μ-Calculus augmented with the step notion leads to an interesting variant.

References

[Bed] Bednarczyk, M. (1988), "Categories of asynchronous systems," Ph.D. thesis, Report 1/88, Dept of Computer Science, Univ. of Sussex.

[Bur] Burgess, J.P. (1984), Basic tense logic, *in* "Handbook of philosophical logic, Vol. II," (D. Gabbay and F. Guenthner, Eds.) pp. 89-133, Reidel.

[BC] Boudol, G., and Castellani, I. (1988), A non-interleaving semantics for CCS based on proved transitions, *Fund. Inform.* **XI**, 433-452.

[DM] Degano, P., and Montanari, U. (1987), Concurrent histories: A basis for observing distributed systems, *J. Comput. Syst. Sci.* **34**, 422-461.

[ES] Emerson, E.A., and Srinivasan, J. (1989), Branching time temporal logic, *LNCS* **354**, 123-172.

[FL] Fischer, M., and Ladner, R. (1981), Propositional dynamic logic of regular programs, *J. Comput. Syst. Sci.* **18**, 194-211.

[Gol] Goldblatt, R. (1987), "Logics of time and computation," Lecture Notes, Centre for Study of Language and Information.

[Har84] Harel, D. (1984), Dynamic logic, *in* "Handbook of philosophical logic, Vol. II," (D. Gabbay and F. Guenthner, Eds.) pp. 497-604, Reidel.

[Har85] Harel, D. (1985), Recurring dominoes: making the highly undecidable highly understandable, *Ann. Disc. Math.* **24**, 51-72.

[HC] Hughes, G.E., and Cresswell, M.J. (1984), "A companion to modal logic," Methuen.

[HM] Hennessy, M., and Milner, R. (1985), Algebraic laws for nondeterminism and concurrency, *J. Assoc. Comput. Mach.* **32**, 137-161.

[Koz] KOZEN, D. (1983), Results on the propositional mu-calculus, *Theoret. Comput. Sci.* **27**, 333-354.

[KP] KOZEN, D., AND PARIKH, R. (1981), An elementary proof of completeness for PDL, *Theoret. Comput. Sci.* **14**, 113-118.

[Krö] KRÖGER, F. (1985), "Temporal logic of programs," Springer-Verlag.

[LP] LEWIS, H.R., AND PAPADIMITRIOU, C.H. (1981), "Elements of the theory of computation," Prentice-Hall.

[LRT] LODAYA, K., RAMANUJAM, R., AND THIAGARAJAN, P.S. (1989), A logic for distributed transition systems, *LNCS* **354**, 508-522.

[Maz] MAZURKIEWICZ, A. (1989), Basic notions of trace theory, *LNCS* **354**, 285-363.

[Mil] MILNER, R. (1989), "Communication and concurrency," Prentice-Hall.

[NPW] NIELSEN, M., PLOTKIN, G., AND WINSKEL, G. (1980), Petri nets, event structures and domains I, *Theoret. Comput. Sci.* **13**, 86-108.

[NRT] NIELSEN, M., ROZENBERG, G., AND THIAGARAJAN, P.S. (1990), Behavioural notions for elementary net systems, *Distr. Comput.* **4**, 45-57.

[Parikh] PARIKH, R. (1988), Decidability and undecidability in distributed transition systems, *in* "A perspective in theoretical computer science — commemorative volume for Gift Siromoney," (R. Narasimhan, Ed.), pp.199-209, World Scientific.

[Park] PARK, D. (1981), Concurrency and automata on infinite sequences, *LNCS* **104**, 167-183.

[Pen] PENCZEK, W. (1992), On undecidability of propositional temporal logics on trace systems, *Inf. Proc. Lett.* **43**, 147-153.

[Pnu] PNUELI, A. (1977), The temporal logic of programs, *in* Proc. 18th Conf. Found. of Comp. Sci., pp. 46-57, IEEE.

[Pra80] PRATT, V. (1980), A near-optimal method for reasoning about action, *J. Comput. Syst. Sci.* **20**, 231-254.

[Pra86] PRATT, V. (1986), Modelling concurrency with partial orders, *Int. J. Parallel Programming* **15**, 33-71.

[Rei] REISIG, W. (1985), "Petri nets – an introduction," Springer-Verlag.

[Shi] SHIELDS, M. (1985), Concurrent machines, *Comput. J.* **28**, 449-465.

[Sta] STARK, E.W. (1989), Concurrent transition systems, *Theoret. Comput. Sci.* **64**, 221-269.

[Sti] STIRLING, C. (1992), Modal and temporal logics, *in* "Handbook of logic in computer science, Vol. 2," (S. Abramsky, D.M. Gabbay and T.S.E. Maibaum, Eds.) pp.477-563, Oxford.

[Thi] THIAGARAJAN, P.S. (1987), Elementary net systems, *LNCS* **254**, 26-59.

[Val] Valiev, M.K. (1993), Π_1^1-universality of some propostional logics of concurrent programs, *Theoret. Comput. Sci.* **119**, 223-232.

[Win] Winskel, G. (1987), Event structures, *LNCS* **255**, 325-392.

[WN] Winskel, G., and Nielsen, M. (1992), "Models for concurrency," Report DAIMI PB-429, Computer Science Dept, Aarhus University.