Theoretical Computer Science 412 (2011) 2072–2092

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Linear logic as a tool for planning under temporal uncertainty

Max Kanovich [a,*], Jacqueline Vauzeilles [b]

[a] Queen Mary University of London, School of Electronic Engineering and Computer Science, Mile End Road, London E1 4NS, UK
[b] LIPN, UMR CNRS 7030, Institut Galilée, Université Paris 13, 99 Av. J.-B. Clément, 93430 Villetaneuse, France

ARTICLE INFO

Keywords:
Linear logic
Artificial intelligence
Planning under uncertainty
Winning strategies
Proofs-as-programs paradigm
Horn linear logic

ABSTRACT

The typical AI problem is that of making a plan of the actions to be performed by a controller so that it could get into a set of *final* situations, if it started with a certain *initial* situation.

The plans, and related winning strategies, happen to be finite in the case of a finite number of states and a finite number of *instant* actions.

The situation becomes much more complex when we deal with planning under *temporal uncertainty* caused by actions with *delayed effects*.

Here we introduce a tree-based formalism to express plans, or winning strategies, in finite state systems in which actions may have *quantitatively delayed effects*. Since the delays are non-deterministic and continuous, we need an infinite branching to display all possible delays. Nevertheless, under reasonable assumptions, we show that infinite winning strategies which may arise in this context can be captured by finite plans.

The above planning problem is specified in logical terms within a Horn fragment of affine logic. Among other things, the advantage of linear logic approach is that we can easily capture 'preemptive/anticipative' plans (in which a new action β may be taken at some moment within the running time of an action α being carried out, in order to be prepared before completion of action α).

In this paper we propose a comprehensive and adequate logical model of strong planning under temporal uncertainty which addresses infinity concerns. In particular, we establish a direct correspondence between linear logic proofs and plans, or winning strategies, for the actions with quantitative delayed effects.

1. Introduction and motivating examples

Linear logic has been shown to be an adequate tool for sorting out planning problems in deterministic as well as in non-deterministic domains [19,20,15].

The main advantage of linear logic approach is a direct and transparent correspondence between proofs for Horn linear logic sequents and plans for AI planning problems. In many cases this allows us to decrease significantly the combinatorial costs associated with searching large spaces [15,16].

The complexity results of [15,16] rely upon the assumption that the actions in question cause only *instant* effects, so that is the duration of the actions equals zero.

In this paper we address the planning problems under *temporal uncertainty* about the effects of actions [2,10] where the time duration does matter. Adding such a 'real time' direction makes the planning problem much more complicated. In particular, plans become winning strategies, and the planning objective is to find a plan that is guaranteed to achieve the goal even within the "worst-case scenario".

* Corresponding author.
E-mail addresses: mik@dcs.qmul.ac.uk (M. Kanovich), jv@lipn.univ-paris13.fr (J. Vauzeilles).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2010.12.027

The aim of the paper is to provide a strict correspondence between proofs and plans even within this temporal setting.

We will illustrate peculiarities and subtleties of the problem with the following simplified version of an example from [10]:

Example 1.1. Assume that a ship is scheduled to leave its original seaport (call it `‘there’`) to be serviced at `‘here’`. The move takes two to five days.

The ship can be serviced either on a normal dock (then she will stay docked two to three days), or on an express dock (then she will stay docked at most one day). But the express dock should be reserved two days in advance, and must be taken exactly two days after the moment the reservation has been made.

The question is to make a plan of actions to *guarantee* that, under any circumstances, the ship will be serviced within seven days? □

The positive answer to Example 1.1 is given, for instance, with the following plan:

l_1: At the initial moment 0, let the ship be bound for `‘here’`. Go to l_2.
l_2: If the ship comes in `‘here’` at some moment t_2 less than 4 time units, go to l_3.
Otherwise, go to l_4 (“If Plan A fails, go to Plan B”).
l_3: At this moment t_2, put her in the normal dock to be serviced. Go to l'_3.
l'_3: Having serviced the ship by some moment t'_2, stop.
(In total, it takes at most $t'_2 \le (t_2+3) \le 7$ days)
l_4: At moment t_1 such that $t_1 = 4$, make a reservation for the express dock. Go to l_5.
l_5: When the ship eventually comes in `‘here’` at some t_2, go to l_6.
l_6: At moment t_3 such that $t_3 = t_1+2$, put her in the express dock to be serviced. Go to l'_6.
l'_6: Having serviced the ship by some moment t'_3, stop.
(In total, it takes at most $t'_3 \le (t_3+1) \le (t_1+3) \le 7$ days) (1)

Remark 1.1. Solving planning problems, we have to address the following issues:

(a) “The guaranteed success, not simple reachability/compatibility”

Following the recommendations of the above plan (1), one can never be punished, since the plan represents a *winning strategy* that envisages *all* possible delays on the road from the initial situation to a final one.

In particular, at every point, the plan provides all preconditions for the corresponding action to be enabled at the given point.

On each of the execution branches, its timestamps form a non-decreasing sequence, with providing compatibility of the time constraints along the branch.

(b) “Preemptive/anticipative actions are vital”

In our example, line l_4 recommends to choose some moment t_1 within the waiting time for the ship’s move from `‘there’` to `‘here’` and to make a reservation for the express dock *in advance* before the ship’s move has been actually completed.

Moreover, we can show that any winning solution to Example 1.1 must include such a ‘preemptive/anticipative’ action: in the case of delays around 6 time units we would have failed if we had allowed the reservation action only after the above *move* action had been fully completed.

(c) “The lock-unlock discipline”

For each action α, the pairs of events “*start an action* α” and “*the action* α *is completed*” form in time a sequence of non-overlapping pairs.

In addition to that, the above plan is *perfect* from the garbage collection point: however the termination step we get, each of the actions involved has been already completed.

2. Real time

We are dealing with the following mathematical model.

A global continuous measurable quantity `time` is assumed in which events occur in irreversible succession from the past through the present to the future.

The *time advance* will be specified with the following `‘Tick’` axioms:

$$\mathtt{T}(t) \vdash \mathtt{T}(t+\varepsilon) \tag{2}$$

where $\mathtt{T}(t)$ denotes *“Time is t”*, and ε is an arbitrary positive real.

Time delays are generally qualified in terms of time intervals such as: “*It takes two to five days.*” Therefore, we will invoke the following basic facts related to time intervals.

As atomic formulas we consider $(t' \le t+h)$, and $(t' < t+h)$, and $(t' = t+h)$, etc. where t and t' are *time variables*, measured in time units, and h is a *real constant*, measured in time units. These atomic formulas may be combined by ‘product’ $\otimes$ and ‘disjunction’ $\oplus$.

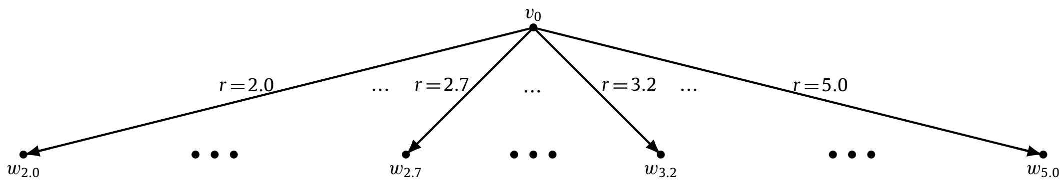

Fig. 1. A step of a winning strategy. We assume that v_0 is labelled by a triple of the form $(s, \mathrm{go}_\alpha, \tau)$, which means that action α is fired in state s at moment τ, and α's performance takes 2–5 time units.

As *axioms of real time* we take basic valid sequents over these combinations of atomic formulas such as:

$$((2 \le \rho \le 3) \otimes (2 \le t_2 < 4) \otimes (t_7 = t_2 + \rho)) \vdash (0 < t_7 \le 7).$$

The 'disjunctive case' may be invoked as:

$$(\rho \le 5) \vdash ((\rho \le 4) \oplus (4 < \rho \le 5)).$$

More generally, as atomic formulas we take

$$(\rho \in \mathcal{E}),$$

where ρ is a *time variable*, measured in time units, and $\mathcal{E}$ is a set taken from a given class of subsets of the 'time scale' `Time`. We will assume that this class is closed under the set union (Fig. 1).

Definition 2.1. As *axioms of real time* we will take:

(a) The valid sequents of the form:

$$((\rho_1 \in \mathcal{E}_1) \otimes (\rho_2 \in \mathcal{E}_2) \otimes \cdots \otimes (\rho_\ell \in \mathcal{E}_\ell) \otimes (t = h(\rho_1, \rho_2, \ldots, \rho_\ell))) \vdash (t \in \mathcal{E}). \tag{3}$$

where $\rho_1, \rho_2, \ldots, \rho_\ell$ are distinct time variables, and $h(\rho_1, \rho_2, \ldots, \rho_\ell)$ is a term over variables $\rho_1, \rho_2, \ldots, \rho_\ell$ and real constants (measured in time units).

(b) The valid sequents of the form ($m \ge 1$):

$$(\rho \in \mathcal{E}) \vdash ((\rho \in \mathcal{E}_1) \oplus (\rho \in \mathcal{E}_2) \oplus \cdots \oplus (\rho \in \mathcal{E}_m)). \tag{4}$$

In fact, the latter reads that $\mathcal{E} \subseteq \mathcal{E}_1 \cup \mathcal{E}_2 \cup \cdots \cup \mathcal{E}_m$.

3. Real-time systems: trajectories

Given an action system, a *trajectory* $\mathcal{F}$ can be conceived of as a mapping

$$\mathcal{F} : \mathtt{Time} \mapsto \mathtt{STATE},$$

showing a possible course of events in the system: $\mathcal{F}(t)$ is the total state observed in the system at moment t.

A common feature of real-time systems is that their laws are not sensitive to the choice of a starting moment, which implies invariance under time translation: $t \mapsto t + \delta$. In its turn, the invariance under time translation provides the *conservation of energy* in the systems. This fact can be understood as a consequence of Noether's theorem [21] that proves that any system will have constant energy, whenever its laws are invariant under shifts in time.

In particular, for such systems, *only finitely many* events may occur within a bounded time interval.

The effect is that, for the systems with a finite number of states, we can confine ourselves to *piecewise constant trajectories* $\mathcal{F}$. Moreover, any event there can be conceived as the *instant change* in the states at some moment t followed by a certain *time advance*, if necessary.

4. Plans. Winning strategies

For AI systems with pure deterministic *instant* actions, a plan $\mathcal{P}$ is defined as a *chain of the actions* leading to the goal [22,10].

Dealing with the actions with *quantitatively delayed effects*, we are involved in a certain *game* against Nature: In order to succeed, we have to respond properly to *each* of the possible quantitative delays on the road from the initial situation to a final one.

Accordingly, we extend their definition to finite *tree-like plans* $\mathcal{P}$, which are supposed to develop (inherently infinite) *winning strategies*: Within such a strategy, each vertex v prescribes the performance of a certain action α for a given state S at a given moment t, the vertex v has an infinite number of the outgoing edges that show all possible delays of displaying the effect caused by the α.

Definition 4.1. Let W be an *initial* state, and $Z_1, Z_2, \ldots, Z_k$ be *final* partial states. Let a task be to make a plan leading from W to either of the final situations within a given time interval A_0 to B_0.

A *winning strategy* $\mathcal{W}$ for this task is defined as a rooted tree all of its branches are finite and in which

(a) Each vertex v is labelled either by a triple of the form $(S, \mathtt{go}_\alpha, \tau)$ or by a triple of the form $(S, \mathtt{end}_\alpha, \tau)$ or by a triple of the form $(S, \mathtt{flash}_\alpha, \tau)$, where
- S is a total state of the system in question;
- for α being an *instant* action, we use $\mathtt{flash}_\alpha$ meaning "α has been performed";
- for α being an action with *delays*, we use $\mathtt{go}_\alpha$ standing for "α is fired", and $\mathtt{end}_\alpha$ meaning "α is completed";
- τ is a *timestamp*, the moment when the corresponding event $\mathtt{go}_\alpha$ or $\mathtt{end}_\alpha$ or $\mathtt{flash}_\alpha$ happens.

Besides,

(a1) The root is labelled by a triple of the form $(W, \mathtt{go}_\alpha, 0)$, or $(W, \mathtt{flash}_\alpha, 0)$, where W is an initial state, 0 is the initial moment.

(a2) For any edge (v, w), where v is labelled by a triple of the form
$$(S, *_\alpha, \tau)$$
and w is labelled by a triple of the form
$$(S', *_\beta, \tau'),$$
the following holds: S' is the result of the event $*_\alpha$ applied to S, and $\tau' \geq \tau$.

Along each of the branches of the tree, these timestamps τ the vertices are labelled by form a non-decreasing sequence of reals.

(a3) Each terminal vertex is labelled by a triple of the form $(S, \mathtt{end}_\alpha, \tau)$ or $(S, \mathtt{flash}_\alpha, \tau)$, such that state S includes one of the final $Z_1, Z_2, \ldots, Z_k$, and $A_0 \leq \tau \leq B_0$.

(b) For any vertex v labelled by a triple of the form
$$(S, \mathtt{go}_\alpha, \tau),$$
its outgoing edges are labelled by nonnegative real numbers r, the possible delays of the effect caused by the action α in v.

Suppose that this α changes some state s into a state s', and α's performance takes a to b time units.

Then, first, S must be of the form[1] $s \otimes \widehat{s}$, which provides the *enabling conditions* for α, and, secondly, for each real r between a and b, there exists an outgoing edge (v, w_r) labelled by the r.

In addition, on each branch starting from the w_r, there exists a vertex u labelled by a triple of the form
$$(s' \otimes \widehat{\widehat{s}}, \mathtt{end}_\alpha, \tau + r)$$
such that no intermediate vertex between v and u is labelled by a triple of the form $(\widehat{S}, \mathtt{go}_\alpha, \widehat{\tau})$, or $(\widehat{S}, \mathtt{end}_\alpha, \widehat{\tau})$.

Thus r is the *time distance* between v (where α has started) and u (where α has been completed), even if the u has happened strictly below w_r: the case where, for instance, some 'preemptive/anticipative' action β has happened at the w_r.[2]

(c) A non-terminal vertex v labelled by a triple of the form $(S, \mathtt{end}_\alpha, \tau)$, or $(S, \mathtt{flash}_\alpha, \tau)$, has exactly one outgoing edge (v, w), this edge remains unlabelled.

Remark 4.1. For the sake of notational uniformity, we will use the following notational conventions.

First, we will label all non-labelled edges of $\mathcal{W}$ with 0.

For a branch b of length ℓ leading from the root to a vertex v, this v can be uniquely identified by the sequence of reals the consecutive edges of b are labelled by:
$$\rho_0, \rho_1, \ldots, \rho_{\ell-1}.$$
In particular, the triple
$$(S, *_\alpha, \tau),$$
the vertex v is labelled by, can be represented as:
$$(S, *_\alpha, t_{\rho_0, \rho_1, \ldots, \rho_{\ell-1}}).$$

Remark 4.2. Within Definition 4.1 we interpret the actions with *quantitatively delayed effects* in terms of a *two-player game*: Controller against Mother Nature.

---

1 Here $A \otimes B$ is conceived of as "A and B co-exist together". See formalities in Section 7.

2 This kind of interference (somewhere in between $\mathtt{go}_\alpha$ and $\mathtt{end}_\alpha$) may have happened only for a *non-instant* action α.

We exclude the case of being delayed *indefinitely*: any action α having been fired is to be eventually completed, so that on each of the branches pairs of the form $\langle \mathtt{go}_\alpha, \mathtt{end}_\alpha \rangle$ must occur in a coherent way.

(1) Controller can perform any move of the form $\mathtt{run}_\beta$. Here, and henceforth, we will use $\mathtt{run}_\beta$ to denote $\mathtt{flash}_\beta$ (for an instant action β) or $\mathtt{go}_\beta$ (for a β with delayed effects).

At a given position w, Controller chooses an action β to be executed, and a moment τ' to start the execution.

Let v be the father of w. Controller may use the following information: the sequence of the triples that label vertices on the branch from the root to the v. Among other things, this information provides the list of actions still running at the moment τ'.

If v is labelled by a triple of the form:

$$(S, *_\alpha, t_{\rho_0,\rho_1,\dots,\rho_{\ell-1}}),$$

then w is labelled by a triple of the form:

$$(S', \mathtt{run}_\beta, t_{\rho_0,\rho_1,\dots,\rho_{\ell-1},\rho_\ell}),$$

where

$$t_{\rho_0,\rho_1,\dots,\rho_{\ell-1},\rho_\ell} = \tau'. \tag{5}$$

(2) Nature can perform only moves of the form: $\mathtt{end}_\alpha$.

At a given position u marked with $\mathtt{end}_\alpha$, Nature responds with a delay ρ to determine the moment τ' of completion of the corresponding α, so that $\tau' = \tau_{\mathtt{go}_\alpha} + \rho$.

More formally, let v be the closest ancestor of u labelled by a triple of the form:

$$(S, \mathtt{go}_\alpha, t_{\rho_0,\rho_1,\dots,\rho_{\ell-1}}).$$

Then u is labelled by a triple of the form:

$$(S', \mathtt{end}_\alpha, t_{\rho_0,\rho_1,\dots,\rho_{\ell-1},\rho_\ell,\dots,\rho_k}),$$

where $t_{\rho_0,\rho_1,\dots,\rho_{\ell-1},\rho_\ell,\dots,\rho_k}$ is defined by the formula:

$$t_{\rho_0,\rho_1,\dots,\rho_{\ell-1},\rho_\ell,\dots,\rho_k} = t_{\rho_0,\rho_1,\dots,\rho_{\ell-1}} + \rho_\ell. \tag{6}$$

5. Winning strategies of bounded height

Though a winning strategy $\mathcal{W}$ does not contain infinite branches, the strategy $\mathcal{W}$ is generally of infinite branching. The effect is that we cannot apply König Lemma to establish a finite bound for its height. Moreover, we can easily make arbitrarily long branches by repeatedly applying, for instance, an action that admits infinitesimal delays.

Nevertheless, under practically reasonable conditions—that any non-instant action takes a positive time, we show how to remove unnecessary repetitions, with resulting in $\mathcal{W}'$ of bounded height.

Theorem 5.1. *Suppose that a system with a finite number of states includes a finite number of instant actions $\beta_1, \dots, \beta_\ell$ and a finite number of actions $\alpha_1, \dots, \alpha_k$ with delayed effects, each α_i's performance takes a_i to b_i time units. Assume that all $a_1, \dots, a_k$ are positive.*

Let a task be to make a plan leading from an initial W to either of the final situations within a given time interval A_0 to B_0, where B_0 is finite.

Then any winning strategy $\mathcal{W}$ for this task can be adjusted to a winning strategy $\mathcal{W}'$ of bounded height.

Proof. Let $\mathcal{B}$ be an arbitrary branch $v_0, \dots, v_N$. The time distance between the timestamps in v_0 and v_N is bounded by B_0.

By K denote the number of vertices on $\mathcal{B}$ that are labelled by non-instant actions.

For a fixed non-instant action α_j, the pairs of vertices labelled by $(S, \mathtt{go}_{\alpha_j}, \tau)$ and $(S', \mathtt{end}_{\alpha_j}, \tau')$, respectively, form a sequence of non-overlapping pairs. Taking into account that $\tau' - \tau \ge a_j$, the number of such pairs does not exceed $\frac{B_0}{\varepsilon}$, where $\varepsilon := \min\{a_1, \dots, a_k\}$.

Hence, the total number K of vertices labelled by non-instant actions does not exceed $2k\frac{B_0}{\varepsilon}$:

$$K \le 2k\frac{B_0}{\varepsilon}.$$

Now we will consider a segment $w_1, \dots, w_l$ on branch $\mathcal{B}$ such that all its vertices are labelled by instant actions only:

$$(S_1, \mathtt{flash}_{\gamma_1}, \tau_1),\ \dots,\ (S_l, \mathtt{flash}_{\gamma_l}, \tau_l).$$

Notice that all these vertices are non-branching.

Assume that $l > \ell \cdot M + 1$, where M is the total number of states in the system.

Then, for some i and j such that $2 \le i < j \le l$, we have: $S_i = S_j$ and $\gamma_i = \gamma_j$.

Now we replace the subsegment

$$(S_{i-1}, \mathtt{flash}_{\gamma_{i-1}}, \tau_{i-1}),\ (S_i, \mathtt{flash}_{\gamma_i}, \tau_i),\ \dots\ (S_j, \mathtt{flash}_{\gamma_j}, \tau_j)$$

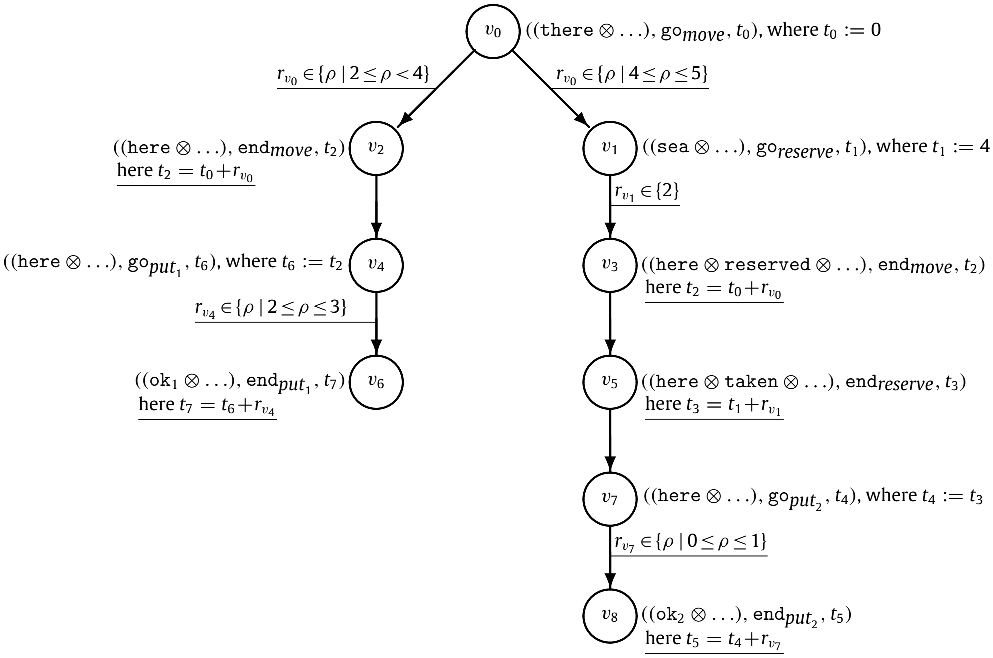

Fig. 2. The winning strategy for Example 1.1. We depict explicitly only parts of the states of the system: $\mathtt{ok}_i$ means "the ship has been serviced on dock$_i$", `reserved` stands for "the express dock is reserved", and `taken` means "the express dock is taken". On the right-hand branch, the events $\mathtt{go}_{move}$ (at moment t_0) and $\mathtt{end}_{move}$ (at moment t_2) are attached to non-adjacent vertices. Nevertheless, by definition: $t_2 = t_0 + r_{v_0}$, in accordance with that r_{v_0} is the *time distance* between these two events.

with the following short subsegment:

$$(S_{i-1}, \mathtt{flash}_{\gamma_{i-1}}, \tau_{i-1}),\ (S_j, \mathtt{flash}_{\gamma_j}, \tau_j).$$

It is readily seen that a "compressed" strategy remains a winning strategy for our planning task.

By repeatedly applying this procedure, we obtain a winning strategy in which the length of each branch does not exceed

$$(2k\frac{B_0}{\varepsilon} + 1)(\ell \cdot M + 1),$$

which provides a finite upper bound for the height of the whole winning strategy. □

6. Plans as a concise representation of winning strategies

Within a winning strategy $\mathcal{W}$ of bounded height, the length of all branches is bounded by a finite number. In the case of a finite number of states and a finite number of actions, this implies certain similarity between branches and subtrees. We explore this effect to develop a *finite 'concise' representation* for the winning strategies, see Lemma 12.1.

Example 6.1. By gluing similar pieces, in Fig. 2 we give a 'concise' winning strategy for Example 1.1, which yields the aforementioned plan (1).

In particular, to the left we group together the delays r_{v_0} between 2 and 4, since the corresponding subtrees happen to be identical but parametrized with the r_{v_0}.

By similar reasons, to the right we group together the delays r_{v_0} between 4 and 5.

Definition 6.1. A *plan* $\mathcal{P}$ is a finite rooted binary tree whose vertices are *labelled commands*, and some of its edges is labelled by time variables representing the *timestamps*.

We will use the following *labelled commands*:

(a) A *command* of the form:

$$\boxed{l\text{: At moment } t_l\text{, start action } \alpha_l;\ \text{ go to } l'.} \tag{7}$$

Here l' is supposed to be a unique child of l.

The above t_l, the *enabling moment of* l, is supposed to be explicitly determined by the *timestamps* labeling some edges on the branch from the root into l.

The outgoing edge (l, l') must be labelled by the t_l.

(b) A *conditional command* of the form:

$$\boxed{\begin{array}{l} l\text{: If action } \alpha_l \text{ is completed at some moment } t_l \text{ less than a given bound } T_l \text{, go to } l'. \\ \quad \text{Otherwise, go to } l''. \end{array}} \tag{8}$$

Here l' and l'' are supposed to be children of l. The bound T_l is supposed to be explicitly determined by the *timestamps* labeling some edges on the branch from the root into l.

(b1) The 'positive' outgoing edge (l, l') must be labelled by the t_l.

(b2) No label is attached to the 'negative' outgoing edge (l, l'').

(c) A particular case of the *conditional command* is of the form:

$$\boxed{l\text{: When action } \alpha_l \text{ is completed at some moment } t_l \text{, go to } l'.} \tag{9}$$

Here l' is supposed to be a unique child of l.

The outgoing edge (l, l') must be labelled by the t_l. (Notice that this t_l, the *enabling moment of* l, is out of our control here, it is provided by Mother Nature)

(d) A *halting command* of the form

$$\boxed{l\text{: Stop.}} \tag{10}$$

Here l is supposed to be a terminal vertex.

Definition 6.2. Let W be an *initial* state, and $Z_1, Z_2, \ldots, Z_k$ be *final* partial states. Let a task be to make a plan leading from W to either of the final situations within a given time interval A_0 to B_0.

We say that a plan $\mathcal{P}$ is a solution to this task if the tree-like strategy $\mathcal{W}$ unfolded according to $\mathcal{P}$ is a well-defined *winning strategy* for this task.

7. Linear logic as a specification language

There are a number of logical formalisms for handling the typical AI problem of *making a plan* of the actions to be performed by a robot so that it could get into a set of *final situations* $\widetilde{Z}$, if it started with a certain *initial situation* W (see, for instance, [22,19,7,17,15,2,10]).

As a logical formalism to specify and sort out planning problems under temporal uncertainty, we use *linear logic* introduced by Girard [11,12] as a resource-sensitive refinement of the traditional logic, see Appendix, Table 2. Allowing *Weakening rule*, we obtain *affine logic*.

In particular, we take advantage of that a linear logic sequent of the 'static' form

$$X \vdash Y$$

can be conceived of as an adequate representation of the *dynamic correlation* between the 'state' X *before* and the 'state' Y *after* the specified event/action has occurred.

Definition 7.1. An *LL theory* T is specified by means of a set of its 'proper axioms' (we denote this set by $\mathtt{Ax}_T$).

An *LL-proof within* T is defined as a standard linear logic derivation tree, excepting that each of its leaves is either a standard axiom of the form $A \vdash A$ or an *instance* of a sequent taken from $\mathtt{Ax}_T$.

Similarly, we define *AL-proofs within* T, with *Weakening rule* being allowed. Here, and henceforth, we will abbreviate: AL = affine logic.

8. Specification of states

Definition 8.1. A *state* s of the system under consideration is represented as an *'elementary product'*:

$$(P_1(s_{1,1}, \ldots, s_{1,k_1}) \otimes P_2(s_{2,1}, \ldots, s_{2,k_2}) \otimes \cdots \otimes P_m(s_{m,1}, \ldots, s_{m,k_m})) \tag{11}$$

where $P_1, P_2, \ldots, P_m$ are predicate symbols, $s_{1,1}, \ldots, s_{1,k_1}, \ldots, s_{m,1}, \ldots, s_{m,k_m}$ are terms.
The fact of being in state s at a given *moment* t is represented as:

$$\mathtt{T}(t) \otimes (P_1(s_{1,1}, \ldots, s_{1,k_1}) \otimes P_2(s_{2,1}, \ldots, s_{2,k_2}) \otimes \cdots \otimes P_m(s_{m,1}, \ldots, s_{m,k_m})) \tag{12}$$

where t is a *time variable*, measured in time units, and $\mathtt{T}(t)$ denotes *"Time is t"*.

9. Specification of actions

Based on dynamic nature of linear logic, we intend to specify actions in the system at hand by means of linear logic sequents.

Suppose that the effect of a given action α fired at a moment t is to change some state s into a state s', and α's performance takes a to b time units.

The first naive attempt is to axiomatize this event in a natural Horn-like way:

$$(\mathtt{T}(t) \otimes s) \vdash \exists\rho\,((a \le \rho \le b) \otimes \mathtt{T}(t+\rho) \otimes s'). \tag{13}$$

The drawback of such a straightforward approach is the lack of capacity to deal directly with *preemptive/anticipative planning*, as in Example 1.1. It should be pointed out that one runs into difficulties with the same problem with other logical and non-logical approaches, like timed transition systems, timed automata, Markov decision processes, etc. (see Section 14).

Nevertheless, linear logic is capable of coping with the problem in a very natural way.

Definition 9.1. To monitor the delayed effect of the given action α, we invoke a specific 'time-guarded' predicate $d_\alpha(x)$, where x is a real number or ∞:

(a) During the performance of α, $d_\alpha(x)$ stands for "The effect of action α will be displayed exactly at moment x";
(b) Whereas $d_\alpha(\infty)$ means that action α is not active for the time being.

(Initially, we set $d_\alpha(\infty)$ for all actions.)

Definition 9.2. Now we will split the *global* 'prolongated' α's performance in two *instantaneous* events as follows:

(a) **"Go$_\alpha$":** α is fired at some moment t, with state s being modified into some intermediate state $\widetilde{s}$, the expecting time delay between a and b time units is recorded with d_α.

We axiomatize this *instant* starting event by a Horn-like sequent:

$$(\mathtt{T}(t) \otimes s \otimes d_\alpha(\infty)) \vdash (\mathtt{T}(t) \otimes \widetilde{s} \otimes \exists\rho\,((a \le \rho \le b) \otimes d_\alpha(t+\rho))). \tag{14}$$

This variable ρ will be referred as a *'delay variable'*.

(b) **"End$_\alpha$":** α is completed at the moment t' recorded by d_α, with $\widetilde{s}$ being modified into the proper s'.

We axiomatize this *instant* finishing event by a Horn-like sequent:

$$(\mathtt{T}(t') \otimes \widetilde{s} \otimes d_\alpha(t')) \vdash (\mathtt{T}(t') \otimes s' \otimes d_\alpha(\infty)). \tag{15}$$

As for an *instant* action β, fired at a moment t, that changes some state s into a state s', we axiomatize its *instant* event **"Flash$_\beta$"** by a simple Horn-like sequent:

$$(\mathtt{T}(t) \otimes s) \vdash (\mathtt{T}(t) \otimes s'). \tag{16}$$

Remark 9.1. One can compare our specification approach with the action representation within temporal planners such as PDDL2.1 [8], LPGP [18], CRIKEY [5].

These systems are based on Nilsson's STRIPS [22,7], which is the base for most of the languages for expressing automated planning problem instances.

According to [22,7], each STRIPS action α is specified in terms of its *precondition Pre*(α), which consists of atomic predicate formulas and/or their negations, and two lists of atomic predicate formulas: *add-list Add*(α) and *delete-list Del*(α).

The action specification is applied to *edit* descriptions of situations instead of being used as an *axiom* in *deducing properties* of situations.

Namely, if α's precondition *Pre*(α) is met, generating a new situation description from an old one is a matter of deleting all the atomic formulas taken from *Del*(α) and adding all the atomic formulas taken from *Add*(α).

Since the classical STRIPS actions are supposed to be *timeless*, the main objective of the temporal planners is to specify actions with the non-zero duration. For these purposes, they introduce something like a *durative action operator da* (see [5]), which is a tuple of the form:

$$da = (C_\vdash, C_\leftrightarrow, C_\dashv, A_\vdash, A_\dashv, D_\vdash, D_\dashv, \Delta)$$

where $C_\vdash$, $C_\leftrightarrow$, and $C_\dashv$ are the sets of atomic predicate formulas that must be true at the start, throughout and the end of the execution, respectively; $A_\vdash$, $A_\dashv$, $D_\vdash$, and $D_\dashv$ specify the add and delete effects at the start and the end of the action, and Δ is the action duration.

Their aim then is to transform duration actions into classical STRIPS actions with the help of certain techniques such as *compressed actions* or specific *instant actions* (see, for instance, [5]).

Though the compression technique has been widely used within temporal planners, this does not provide either completeness or soundness (see [6,5]).

Although CRIKEY uses an alternative approach [5], CRIKEY is not a complete planner either: the planner is not guaranteed to find a solution to certain *solvable* planning problems expressible in its action language (see details in [6,5]).

10. Example 1.1: Specification

According to what has been said, the actions in Example 1.1 are axiomatized as follows:

(a) The *move* action: "The ship is bound for `'here'`, which takes two to five days," invokes its 'time-guarded' predicate $d_m(x)$:

(a1) The starting event $\mathtt{go}_{move}$ occurred at moment t (we abbreviate it as $\mathtt{go}_{move}@t$) is specified as[3]:

$$(\mathtt{T}(t) \otimes \mathtt{there} \otimes d_m(\infty)) \vdash (\mathtt{T}(t) \otimes \mathtt{sea} \otimes \exists\rho\, ((2 \le \rho \le 5) \otimes d_m(t+\rho))). \tag{17}$$

(a2) The finishing event $\mathtt{end}_{move}$ at moment t (abbreviated as $\mathtt{end}_{move}@t$) is axiomatized as:

$$(\mathtt{T}(t) \otimes \mathtt{sea} \otimes d_m(t)) \vdash (\mathtt{T}(t) \otimes \mathtt{here} \otimes d_m(\infty)). \tag{18}$$

(b) The put_1 action: "The ship is serviced on the normal dock, where she will stay docked two to three days," invokes its 'time-guarded' predicate $d_1(x)$:

(b1) $\mathtt{go}_{put_1}@t$ is represented as:

$$(\mathtt{T}(t) \otimes \mathtt{here} \otimes d_1(\infty)) \vdash (\mathtt{T}(t) \otimes \mathtt{dock}_1 \otimes \exists\rho\, ((2 \le \rho \le 3) \otimes d_1(t+\rho))) \tag{19}$$

(b2) $\mathtt{end}_{put_1}@t$ is specified as:

$$(\mathtt{T}(t) \otimes \mathtt{dock}_1 \otimes d_1(t)) \vdash (\mathtt{T}(t) \otimes \mathtt{ok}_1 \otimes d_1(\infty)). \tag{20}$$

(c) The *reserve* action: "The express dock is reserved in advance," is specified with invoking its 'time-guarded' predicate $d_r(x)$:

(c1) $\mathtt{go}_{reserve}@t$ is axiomatized as:

$$(\mathtt{T}(t) \otimes d_r(\infty)) \vdash (\mathtt{T}(t) \otimes d_r(t+2)) \tag{21}$$

(c2) $\mathtt{end}_{reserve}@t$ is axiomatized as:

$$(\mathtt{T}(t) \otimes d_r(t)) \vdash (\mathtt{T}(t) \otimes d_r(\infty)). \tag{22}$$

(d) The put_2 action: "The ship is serviced on the express dock, where she will stay docked at most one day," invokes its 'time-guarded' predicate $d_2(x)$:

(d1) $\mathtt{end}_{reserve}$-$\mathtt{go}_{put_2}@t$ is represented as[4]:

$$(\mathtt{T}(t) \otimes \mathtt{here} \otimes d_r(t) \otimes d_2(\infty)) \vdash (\mathtt{T}(t) \otimes \mathtt{dock}_2 \otimes d_r(\infty) \otimes \exists\rho\, ((0 \le \rho \le 1) \otimes d_2(t+\rho))) \tag{23}$$

(d2) $\mathtt{end}_{put_2}@t$ is specified as:

$$(\mathtt{T}(t) \otimes \mathtt{dock}_2 \otimes d_2(t)) \vdash (\mathtt{T}(t) \otimes \mathtt{ok}_2 \otimes d_2(\infty)). \tag{24}$$

The *initial situation* W in Example 1.1 is specified as:

$$(\mathtt{T}(0) \otimes \mathtt{there} \otimes d(\infty, \infty, \infty, \infty)) \tag{25}$$

where, for the sake of brevity:

$$d(x, y, z, u) := d_m(x) \otimes d_1(y) \otimes d_2(z) \otimes d_r(u). \tag{26}$$

The *planning goal* is to get into the set of *final situations* $\widetilde{Z}$ represented as:

$$\widetilde{Z} := \exists t'\, ((0 \le t' \le 7) \otimes \mathtt{T}(t') \otimes (\mathtt{ok}_1 \oplus \mathtt{ok}_2) \otimes d(\infty, \infty, \infty, \infty)) \tag{27}$$

(with $d(\infty, \infty, \infty, \infty)$ we emphasize that we are looking for *perfect plans* where each of the actions involved must be completed up to the final moment t').

---

3 Here `sea` means "nowhere", the *complement* to all others possible states of the ship.

4 To contract the number of states, and to show flexibility of our formalism, we combine $\mathtt{end}_{reserve}$ and $\mathtt{go}_{put_2}$, with including the end of the reservation act as a precondition to enable the act of putting the ship in the express dock.

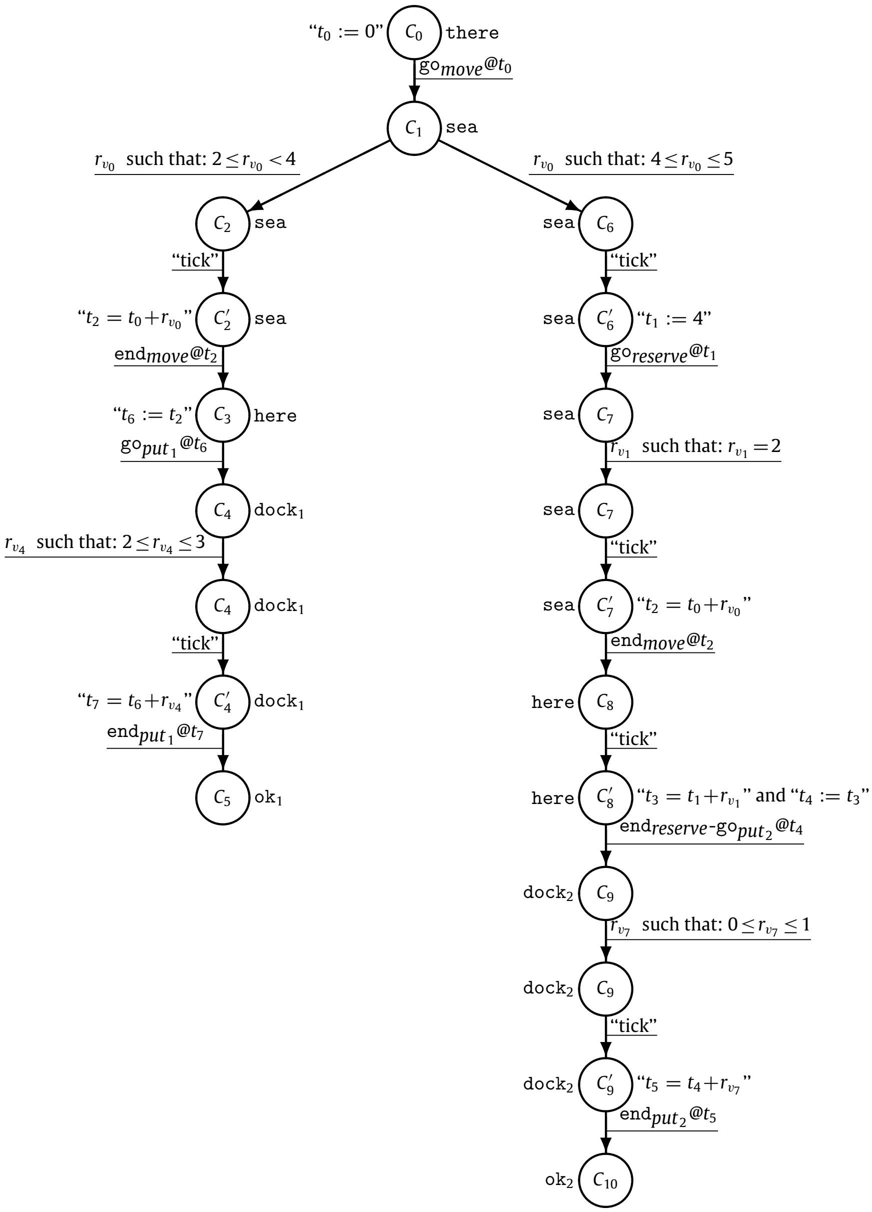

Fig. 3. The game scenario developed in accordance with our winning strategy in Fig. 2. The extended states C_i are given in Example 11.1.

11. Example 1.1: Winning strategies $\Longleftrightarrow$ LL proofs

Given a planning task, we show how to convert its winning strategies (of a finite height) into linear logic proofs for the sequent specifying the task.

Example 11.1. By going into smaller details, we transform the winning strategy in Fig. 2 in a *game scenario* for Example 1.1.

The detailed game scenario is shown in Fig. 3, in which the edges are labelled either by the $\mathtt{go}$/$\mathtt{end}$ events of the corresponding actions, or by time delays r of their effects. Each of the vertices is labelled by an 'extended' state C_i in a style of (33), which contains information about:

- the current moment τ,
- the state S of the system,
- the status of 'time-guarded' predicates $d_\alpha(x)$,
- delays r_v, and timestamps involved by the moment τ.

We use the following 'extended' states C_i within Fig. 3:

$$\dfrac{\mathtt{T}(0)\vdash \mathtt{T}(t_2) \quad \dfrac{\mathtt{end}_{move}@t_2 \quad \dfrac{\mathtt{go}_{put_1}@t_2 \quad \dfrac{\mathtt{T}(t_2)\vdash \mathtt{T}(t_7)\quad \dfrac{\mathtt{end}_{put_1}@t_7 \quad \dfrac{\text{“See (28)”}}{C_5\vdash\widetilde{Z}}}{C_4'\vdash\widetilde{Z}}(\text{ax-cut})}{C_4\vdash \widetilde{Z}}(\text{ax-cut})}{C_3\vdash\widetilde{Z}}(\exists\text{-ax-cut})}{C_2'\vdash\widetilde{Z}}(\text{ax-cut})}{C_2\vdash\widetilde{Z}}(\text{ax-cut})$$

Fig. 4. The E-Horn proof for $C_2 \vdash \widetilde{Z}$; cf. the left branch in the tree in Fig. 3.

$$\dfrac{\mathtt{go}_{move}@0 \quad \dfrac{(2\le r_{v_0}\le 5)\vdash((2\le r_{v_0}<4)\oplus(4\le r_{v_0}\le 5)) \quad \dfrac{\text{“See Fig. 4”}}{C_2\vdash\widetilde{Z}} \quad \dfrac{\text{“Like Fig. 4”}}{C_6\vdash\widetilde{Z}}}{C_1\vdash\widetilde{Z}}(\oplus\text{-ax-cut})}{C_0\vdash\widetilde{Z}}(\exists\text{-ax-cut})$$

Fig. 5. The E-Horn proof for $C_0 \vdash \widetilde{Z}$; cf. the tree in Fig. 3.

(1) $C_0 = (\mathtt{T}(0) \otimes \mathtt{there} \otimes d(\infty, \infty, \infty, \infty))$
(2) $C_1 = (\mathtt{T}(0) \otimes \mathtt{sea} \otimes (2 \le r_{v_0} \le 5) \otimes d(0+r_{v_0}, \infty, \infty, \infty))$
(3) $C_2 = (\mathtt{T}(0) \otimes \mathtt{sea} \otimes (2 \le r_{v_0} < 4) \otimes d(r_{v_0}, \infty, \infty, \infty))$
(4) $C_6 = (\mathtt{T}(0) \otimes \mathtt{sea} \otimes (4 \le r_{v_0} \le 5) \otimes d(r_{v_0}, \infty, \infty, \infty))$ Notice that $C_1 \equiv (C_2 \oplus C_6)$.
(5) $C_2' = (\mathtt{T}(t_2) \otimes \mathtt{sea} \otimes (t_2 = r_{v_0}) \otimes (2 \le r_{v_0} < 4) \otimes d(t_2, \infty, \infty, \infty))$
(6) $C_3 = (\mathtt{T}(t_2) \otimes \mathtt{here} \otimes (t_2 = r_{v_0}) \otimes (2 \le r_{v_0} < 4) \otimes d(\infty, \infty, \infty, \infty))$
(7) $C_4 = (\mathtt{T}(t_2) \otimes \mathtt{dock}_1 \otimes (2 \le r_{v_4} \le 3) \otimes (t_2 = r_{v_0}) \otimes (2 \le r_{v_0} < 4) \otimes d(\infty, t_2+r_{v_4}, \infty, \infty))$
(8) $C_4' = (\mathtt{T}(t_7) \otimes \mathtt{dock}_1 \otimes (t_7 = t_2+r_{v_4}) \otimes (2 \le r_{v_4} \le 3) \otimes (t_2 = r_{v_0}) \otimes (2 \le r_{v_0} < 4) \otimes d(\infty, t_7, \infty, \infty))$
(9) $C_5 = (\mathtt{T}(t_7) \otimes \mathtt{ok}_1 \otimes (t_7 = t_2+r_{v_4}) \otimes (2 \le r_{v_4} \le 3) \otimes (t_2 = r_{v_0}) \otimes (2 \le r_{v_0} < 4) \otimes d(\infty, \infty, \infty, \infty))$

Notice that $C_5 \vdash \widetilde{Z}$ is derivable by rules taken from the system in Definition 13.1:

$$\dfrac{((t_7 = t_2+r_{v_4}) \otimes (2 \le r_{v_4} \le 3) \otimes (t_2 = r_{v_0}) \otimes (2 \le r_{v_0} < 4)) \vdash (0 \le t_7 \le 7) \qquad \dfrac{}{\widetilde{C}_5 \vdash \widetilde{Z}}(\text{E-axiom})}{C_5 \vdash \widetilde{Z}}(\text{ax-cut}). \tag{28}$$

Here $\widetilde{C}_5 = (\mathtt{T}(t_7) \otimes \mathtt{ok}_1 \otimes (0 \le t_7 \le 7) \otimes d(\infty, \infty, \infty, \infty))$
and $\widetilde{Z} := \exists t' \, ((0 \le t' \le 7) \otimes \mathtt{T}(t') \otimes (\mathtt{ok}_1 \oplus \mathtt{ok}_2) \otimes d(\infty, \infty, \infty, \infty))$

(10) $C_6' = (\mathtt{T}(t_1) \otimes \mathtt{sea} \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(r_{v_0}, \infty, \infty, \infty))$
(11) $C_7 = (\mathtt{T}(t_1) \otimes \mathtt{sea} \otimes (r_{v_1} = 2) \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(r_{v_0}, \infty, \infty, t_1+r_{v_1}))$
(12) $C_7' = (\mathtt{T}(t_2) \otimes \mathtt{sea} \otimes (t_2 = r_{v_0}) \otimes (r_{v_1} = 2) \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(t_2, \infty, \infty, 6))$
(13) $C_8 = (\mathtt{T}(t_2) \otimes \mathtt{here} \otimes (t_2 = r_{v_0}) \otimes (r_{v_1} = 2) \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(\infty, \infty, \infty, 6))$
(14) $C_8' = (\mathtt{T}(t_4) \otimes \mathtt{here} \otimes (t_4 = 6) \otimes (t_2 = r_{v_0}) \otimes (r_{v_1} = 2) \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(\infty, \infty, \infty, t_4))$
(15) $C_9 = (\mathtt{T}(t_4) \otimes \mathtt{dock}_2 \otimes (0 \le r_{v_7} \le 1) \otimes (t_4 = 6) \otimes (t_2 = r_{v_0}) \otimes (r_{v_1} = 2) \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(\infty, \infty, t_4+r_{v_7}, \infty))$
(16) $C_9' = (\mathtt{T}(t_5) \otimes \mathtt{dock}_2 \otimes (t_5 = t_4+r_{v_7}) \otimes (0 \le r_{v_7} \le 1) \otimes (t_4 = 6) \otimes (t_2 = r_{v_0}) \otimes (r_{v_1} = 2) \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(\infty, \infty, t_5, \infty))$
(17) $C_{10} = (\mathtt{T}(t_5) \otimes \mathtt{ok}_2 \otimes (t_5 = t_4+r_{v_7}) \otimes (0 \le r_{v_7} \le 1) \otimes (t_4 = 6) \otimes (t_2 = r_{v_0}) \otimes (r_{v_1} = 2) \otimes (t_1 = 4) \otimes (4 \le r_{v_0} \le 5) \otimes d(\infty, \infty, \infty, \infty))$.

Notice that $C_{10} \vdash \widetilde{Z}$ is also derivable by rules taken from the system in Definition 13.1.

Furthermore, in accordance with a general recipe of Theorem 12.1, which recommends, in particular, a proof construct of the form (34):

$$\dfrac{\mathtt{go}_\alpha@\tau \quad \dfrac{(a \le \rho_\ell \le b) \vdash ((\rho_\ell \in \mathcal{E}_1) \oplus (\rho_\ell \in \mathcal{E}_2)) \quad \dfrac{\mathtt{T}(\tau) \vdash \mathtt{T}(\tau_1) \quad C_{w_1} \vdash \widetilde{Z}}{\widehat{C}_{w_1} \vdash \widetilde{Z}} \quad \dfrac{\mathtt{T}(\tau) \vdash \mathtt{T}(\tau_2) \quad C_{w_2} \vdash \widetilde{Z}}{\widehat{C}_{w_2} \vdash \widetilde{Z}}(\text{ax-cut})}{\widehat{C}_w \vdash \widetilde{Z}}(\oplus\text{-ax-cut})}{C_v \vdash \widetilde{Z}}(\exists\text{-ax-cut})$$

we can easily convert the game scenario in Fig. 3 into a proof for the task sequent (see Figs. 4 and 5). □

Example 11.2. In its turn, the plan proposed in Example 1.1 can be directly extracted from the proof given in Fig. 5 for the 'task sequent':

$$(\mathtt{T}(0) \otimes \mathtt{there} \otimes d(\infty, \infty, \infty, \infty)) \vdash \widetilde{Z}$$

where $\widetilde{Z} := \exists t' \, ((0 \le t' \le 7) \otimes \mathtt{T}(t') \otimes (\mathtt{ok}_1 \oplus \mathtt{ok}_2) \otimes d(\infty, \infty, \infty, \infty))$ □

12. LL proofs $\Longleftrightarrow$ winning strategies

Given a system with a finite set of actions with delayed effects, let W be an *initial* state, and $Z_1, Z_2, \ldots, Z_k$ be *final* partial states. Let `Task` be to make a plan leading from W to either of the final situations within a given time interval A_0 to B_0.

We encode this `Task` as a sequent of the form:

$$(\mathrm{T}(0) \otimes W \otimes \bigotimes_{\alpha} d_\alpha(\infty)) \vdash \exists t' \, ((A_0 \le t' \le B_0) \otimes \mathrm{T}(t') \otimes \overline{Z}^{\oplus} \otimes \bigotimes_{\alpha} d_\alpha(\infty)) \tag{29}$$

where $\overline{Z}^{\oplus} = (Z_1 \oplus Z_2 \oplus \cdots \oplus Z_k)$, and each action α is supplied with its 'time-guarded' predicate $d_\alpha(x)$.

Lemma 12.1. *Any winning strategy $\mathcal{W}$ of bounded height can be represented in a finite concise form $\widetilde{\mathcal{W}}$.*

Proof. Let $\mathcal{W}$ be a winning strategy of height h.

First step. We extract its *timeless skeleton* $\mathcal{W}'$ by removing all timestamps τ from the labels on the vertices and all 'delay' labels r on the edges. Within $\mathcal{W}'$, all edges have no labels, and each vertex v is labelled by a pair of the form

$$(S, *_\alpha).$$

Now running from the leaves of $\mathcal{W}'$ to its root, we construct a finite version, call it $\widetilde{\mathcal{W}}$, that represents the whole $\mathcal{W}'$. Any vertex $\widetilde{v}$ in $\widetilde{\mathcal{W}}$ will be formed as a set of vertices taken from $\mathcal{W}'$ in the following way:

(a) Suppose the terminal vertices $w_1, w_2, \ldots$ are sons of the same vertex v.
Since the number of states is finite and the number of actions is finite, the number of $w_1, w_2, \ldots$ with different labels must be finite. Then we glue together the terminal vertices with identical labels, resulting in a finite number of equivalence classes $\widetilde{w}^1, \widetilde{w}^2, \ldots, \widetilde{w}^k$. We will consider these vertices as the sons of a vertex $\widetilde{v}$, an exact copy of v.

(b) Let a vertex v in $\mathcal{W}'$ have subtrees $T_1, T_2, \ldots,$ such that their finite representatives $\widetilde{T}_1, \widetilde{T}_2, \ldots$ have been already constructed.
Since the number of states is finite and the number of actions is finite, the number of $\widetilde{T}_1, \widetilde{T}_2, \ldots$ with different labels must be finite. Then we glue together the identical subtrees with identical labels, resulting in a finite number of $\widetilde{T}^1, \widetilde{T}^2, \ldots, \widetilde{T}^k$. We will consider these trees as the subtrees of a vertex $\widetilde{v}$, an exact copy of v.

Second step. Running from the root of $\widetilde{\mathcal{W}}$, to its leaves, we restore the timed information but in a 'parametrized' form, resulting in the finite concise representation of the original $\mathcal{W}$. (Cf. Example 6.1.)

We will use notational conventions from Remarks 4.1 and 4.2.

Starting from the root to the leaves, for each level ℓ we introduce a parameter ρ_ℓ.

Assume a vertex $\widetilde{v}$ on level ℓ be labelled by a pair of the form $(S, *_\alpha)$.

We will expand the pair to a triple of the form

$$(S, *_\alpha, t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1}}),$$

where $t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1}}$ is a function over parameters $\rho_0, \rho_1, \ldots, \rho_{\ell-1}$. We will label an outgoing edge $(\widetilde{v}, \widetilde{w})$ with an expression of the form

$$\text{“}\rho_\ell \in \mathcal{D}_{\widetilde{w};\rho_0,\rho_1,\ldots,\rho_{\ell-1}}\text{”},$$

where $\mathcal{D}_{\widetilde{w};\rho_0,\rho_1,\ldots,\rho_{\ell-1}}$ is a set of reals parametrized with $\rho_0, \rho_1, \ldots, \rho_{\ell-1}$.

(a) The root of $\widetilde{\mathcal{W}}$, which is on level 0, is labelled by a pair of the form $(W, \mathtt{run}_\alpha)$. We expand this pair to the triple

$$(W, \mathtt{run}_\alpha, 0).$$

(b) Given $\widetilde{w}$ and $\rho_0, \rho_1, \ldots, \rho_{\ell-1}$, we introduce $\mathcal{D}_{\widetilde{w};\rho_0,\rho_1,\ldots,\rho_{\ell-1}}$ by induction on ℓ.
Let $\widetilde{v}_0, \widetilde{v}_1, \widetilde{v}_2, \ldots, \widetilde{v}_\ell, \widetilde{v}_{\ell+1}$ be the branch that leads from the root to $\widetilde{w}$. For any sequence of reals $r_0, r_1, r_2, \ldots, r_{\ell-1}$ taken from $\mathcal{D}_{\widetilde{v}_1;}, \mathcal{D}_{\widetilde{v}_2;r_0}, \mathcal{D}_{\widetilde{v}_3;r_0,r_1}, \ldots, \mathcal{D}_{\widetilde{v}_\ell;r_0,r_1,\ldots,r_{\ell-2}}$, respectively, we find $v \in \widetilde{v}_\ell$ that is uniquely identified in the original $\mathcal{W}$ by the sequence of edge labels $r_0, r_1, r_2, \ldots, r_{\ell-1}$, and define $\mathcal{D}_{\widetilde{w};r_0,r_1,\ldots,r_{\ell-1}}$ as:

$$\mathcal{D}_{\widetilde{w};r_0,r_1,\ldots,r_{\ell-1}} := \{r \mid \text{for some } w \in \widetilde{w}, \text{ the edge } (v, w) \text{ in } \mathcal{W} \text{ is labelled by } r\}. \tag{30}$$

(c) Suppose a vertex $\widetilde{w}$ on level $\ell+1$ is labelled by a pair of the form $(S', \mathtt{run}_\beta)$.
We expand this pair to the triple

$$(S', \mathtt{run}_\beta, t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1},\rho_\ell}),$$

where $t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1},\rho_\ell}$ is defined as the following function over $\rho_0, \rho_1, \ldots, \rho_{\ell-1}, \rho_\ell$.
Let $\widetilde{v}_0, \widetilde{v}_1, \widetilde{v}_2, \ldots, \widetilde{v}_\ell, \widetilde{v}_{\ell+1}$ be the branch that leads from the root to $\widetilde{w}$.
For any sequence of reals $r_0, r_1, r_2, \ldots, r_{\ell-1}, r_\ell$ taken from $\mathcal{D}_{\widetilde{v}_1;}, \mathcal{D}_{\widetilde{v}_2;r_0}, \mathcal{D}_{\widetilde{v}_3;r_0,r_1}, \ldots, \mathcal{D}_{\widetilde{v}_\ell;r_0,r_1,\ldots,r_{\ell-2}}, \mathcal{D}_{\widetilde{v}_{\ell+1};r_0,r_1,\ldots,r_{\ell-2},r_{\ell-1}}$, respectively, we find the vertex $w \in \widetilde{w}$ that is uniquely identified in the original $\mathcal{W}$ by the sequence of edge labels $r_0, r_1, r_2, \ldots, r_{\ell-1}, r_\ell$. This w is labelled by a triple of the form $(S', \mathtt{run}_\beta, \tau')$.
Then we set:

$$t_{r_0,r_1,\ldots,r_{\ell-1},r_\ell} = \tau'. \tag{31}$$

(d) Suppose a vertex $\widetilde{v}$ on level ℓ has been already labelled by a triple of the form

$$(S, \mathtt{go}_\alpha, t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1}}),$$

and $\widetilde{u}$ is a descendant of $\widetilde{v}$ labelled by a pair of the form $(S', \mathtt{end}_\alpha)$, such that no intermediate vertex between $\widetilde{v}$ and $\widetilde{u}$ is labelled by $(S'', \mathtt{end}_\alpha)$.

Then we label $\widetilde{u}$ with a triple of the form

$$(S', \mathtt{end}_\alpha, t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1},\rho_\ell,\ldots,\rho_k}),$$

where $t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1},\rho_\ell,\ldots,\rho_k}$ is defined by the formula:

$$t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1},\rho_\ell,\ldots,\rho_k} = t_{\rho_0,\rho_1,\ldots,\rho_{\ell-1}} + \rho_\ell. \tag{32}$$

By construction, any branch of length ℓ in the original $\mathcal{W}$ is correctly represented within $\widetilde{\mathcal{W}}$ by taking the corresponding values $r_0, r_1, r_2, \ldots, r_{\ell-1}$ for parameters $\rho_0, \rho_1, \rho_2, \ldots, \rho_{\ell-1}$. Hence, this finite concise $\widetilde{\mathcal{W}}$ is a correct representation for the original $\mathcal{W}$. $\square$

Theorem 12.1 (*Soundness and Completeness*). *Given a system with a finite set of actions α with delayed effects, let W be an* initial *state, and $Z_1, Z_2, \ldots, Z_k$ be* final *partial states. Starting from W, the task* Task *is to achieve either of the final situations within a given time interval A_0 to B_0.*

Let Th *be an affine logic theory that includes as its proper axioms the Horn-like specifications of all actions α, and the appropriate axioms of real time (see Definition 2.1).*

Then a sequent of the form (29):

$$\left(\mathrm{T}(0) \otimes W \otimes \bigotimes_{\alpha} d_\alpha(\infty)\right) \vdash \exists t' \left((A_0 \le t' \le B_0) \otimes \mathrm{T}(t') \otimes \overline{Z}^{\oplus} \otimes \bigotimes_{\alpha} d_\alpha(\infty)\right)$$

is provable in Th *if and only if there exists a winning strategy $\mathcal{W}$ of bounded height for* Task.

Moreover, there is a direct correspondence between Th*-proofs for this sequent and winning strategies (in a finite concise form) that are solution to* Task.

Proof.

(A) **"Strategies $\Longrightarrow$ Proofs".**

We assume that $\mathcal{W}$ is represented in a finite concise form by construction in Lemma 12.1.

With each vertex u at level ℓ labelled by $(S, *_\alpha, \tau)$, we associate an 'extended' state C_u:

$$C_u = \left(\mathrm{T}(\tau) \otimes S \otimes \bigotimes_{\alpha} d_\alpha(x_\alpha) \otimes \bigotimes_{i=0}^{\ell-1} (\rho_i \in \mathcal{D}_i)\right), \tag{33}$$

that contains information about:

- the current moment τ,
- the state S of the system,
- the status of 'time-guarded' predicates $d_\alpha(x)$,
- delays ρ_i involved at the current moment τ (cf. Example 11.1).

Running from the leaves of this concise version to its root, by induction we assemble a Th-derivation for each of the $C_u \vdash \widetilde{Z}$, where $\widetilde{Z}$ is the right-hand side of (29).

We will consider here a representative case, which captures main induction subtleties.

Suppose a vertex v on level ℓ is labelled by a triple of the form $(S, \mathtt{go}_\alpha, \tau)$, where action α, fired at moment τ, changes the state S into a state S', and the expecting time delay is ρ_ℓ between a and b (see Definition 9.2).

Suppose that v has exactly two sons: w_1 and w_2, labelled by $(S', *_{\beta_1}, \tau_1)$ and $(S', *_{\beta_2}, \tau_2)$, respectively. The edges (v, w_1) and (v, w_2) are labelled by "$\rho_\ell \in \mathcal{E}_1$" and "$\rho_\ell \in \mathcal{E}_2$", respectively, which means that $\mathcal{E}_1 \cup \mathcal{E}_2$ contains all possible delays of action α.

Let $v_0, v_1, v_2, \ldots, v_\ell$ be the branch in $\mathcal{W}$ that leads from the root to v, and each edge (v_i, v_{i+1}) be labelled by "$\rho_i \in \mathcal{D}_i$".

With our vertices we associate the following 'extended' states:

- $C_v = (\mathrm{T}(\tau) \otimes S \otimes d_\alpha(\infty) \otimes \bigotimes_{\beta\neq\alpha} d_\beta(x_\beta) \otimes \bigotimes_{i=0}^{\ell-1} (\rho_i \in \mathcal{D}_i))$
- $C_{w_1} = (\mathrm{T}(\tau_1) \otimes S' \otimes (d_\alpha(\tau+\rho_\ell) \otimes (\rho_\ell \in \mathcal{E}_1)) \otimes \bigotimes_{\beta\neq\alpha} d_\beta(x_\beta) \otimes \bigotimes_{i=0}^{\ell-1} (\rho_i \in \mathcal{D}_i))$
- $C_{w_2} = (\mathrm{T}(\tau_2) \otimes S' \otimes (d_\alpha(\tau+\rho_\ell) \otimes (\rho_\ell \in \mathcal{E}_2)) \otimes \bigotimes_{\beta\neq\alpha} d_\beta(x_\beta) \otimes \bigotimes_{i=0}^{\ell-1} (\rho_i \in \mathcal{D}_i))$

where $\bigotimes_{\beta\neq\alpha} d_\beta(x_\beta)$ represents the status of 'time-guarded' predicates other than d_α (the behaviour of d_α is explained in Definition 9.1).

Let $\mathrm{go}_\alpha@\tau$ denote the 'axiom' (14) where t is taken as τ.

Then we can derive in linear logic (with the rules from the system in Definition 13.1):

$$\dfrac{\mathrm{go}_\alpha@\tau \qquad \dfrac{(a\leq\rho_\ell\leq b)\vdash((\rho_\ell\in\mathcal{E}_1)\oplus(\rho_\ell\in\mathcal{E}_2)) \qquad \dfrac{\mathrm{T}(\tau)\vdash\mathrm{T}(\tau_1)\quad C_{w_1}\vdash\widetilde{Z}}{\widehat{C}_{w_1}\vdash\widetilde{Z}} \qquad \dfrac{\mathrm{T}(\tau)\vdash\mathrm{T}(\tau_2)\quad C_{w_2}\vdash\widetilde{Z}}{\widehat{C}_{w_2}\vdash\widetilde{Z}}\,(\text{ax-cut})}{\widehat{C}_w\vdash\widetilde{Z}}\,(\oplus\text{-ax-cut})}{C_v\vdash\widetilde{Z}}\,(\exists\text{-ax-cut}) \tag{34}$$

Here the auxiliary 'states' $\widehat{C}$ are introduced as:

- $\widehat{C}_{w_1} = (\mathrm{T}(\tau)\otimes S'\otimes(d_\alpha(\tau+\rho_\ell)\otimes(\rho_\ell\in\mathcal{E}_1))\otimes\bigotimes_{\beta\neq\alpha}d_\beta(x_\beta)\otimes\bigotimes_{i=0}^{\ell-1}(\rho_i\in\mathcal{D}_i))$
- $\widehat{C}_{w_2} = (\mathrm{T}(\tau)\otimes S'\otimes(d_\alpha(\tau+\rho_\ell)\otimes(\rho_\ell\in\mathcal{E}_2))\otimes\bigotimes_{\beta\neq\alpha}d_\beta(x_\beta)\otimes\bigotimes_{i=0}^{\ell-1}(\rho_i\in\mathcal{D}_i))$
- $\widehat{C}_{w} = (\mathrm{T}(\tau)\otimes S'\otimes(d_\alpha(\tau+\rho_\ell)\otimes(a\leq\rho_\ell\leq b)\otimes\bigotimes_{\beta\neq\alpha}d_\beta(x_\beta)\otimes\bigotimes_{i=0}^{\ell-1}(\rho_i\in\mathcal{D}_i))$.

The fact that all possible delays ρ_ℓ of action α belong to $\mathcal{E}_1\cup\mathcal{E}_2$ is expressed as the following 'axiom of real time' (see Definition 2.1):

$$(a\leq\rho_\ell\leq b)\vdash((\rho_\ell\in\mathcal{E}_1)\oplus(\rho_\ell\in\mathcal{E}_2))\,.$$

By inductive hypothesis, $C_{w_1}\vdash\widetilde{Z}$ and $C_{w_2}\vdash\widetilde{Z}$ are provable in Th.

Therefore, we can conclude that $C_v\vdash\widetilde{Z}$ is also provable in Th, which justifies our bottom-up induction (cf. Figs. 4 and 5).

Lastly, the task sequent (29) is provable in Th, since the 'extended' state C_{v_0}, associated with the root v_0, happens to be the left-hand side of (29):

$$C_{v_0} = \left(\mathrm{T}(0)\otimes W\otimes\bigotimes_\alpha d_\alpha(\infty)\right).$$

(B) **"Proofs $\Longrightarrow$ Strategies".**

The main idea is as follows (cf. [13–16]).

Given a Th-proof proof for the sequent in question, which is of a specific Horn-like form (29):

$$\left(\mathrm{T}(0)\otimes W\otimes\bigotimes_\alpha d_\alpha(\infty)\right)\vdash\exists t'\left((A_0\leq t'\leq B_0)\otimes\mathrm{T}(t')\otimes\overline{Z}^{\oplus}\otimes\bigotimes_\alpha d_\alpha(\infty)\right),$$

with the help of Lemma 13.1 we translate it into a pure affine logic proof D' for a sequent of the form (40):

$$\left(\mathrm{T}(0)\otimes W\otimes\bigotimes_\alpha d_\alpha(\infty)\right),\ !\Delta\vdash\exists t'\left((A_0\leq t'\leq B_0)\otimes\mathrm{T}(t')\otimes\overline{Z}^{\oplus}\otimes\bigotimes_\alpha d_\alpha(\infty)\right),$$

so that within D' we apply only Horn-like rules from Table 1.

Then, running from the leaves of D' to its root, by induction we assemble a solution to Task in the form of a winning strategy in a finite concise form.

We will consider here the most complicated case, in which the rule ($\exists$-**H**) is applied. Recall that the formula $(X\multimap(Y\otimes\exists\rho\,U(\rho)))$ introduced by ($\exists$-**H**) is to be an instance of the LL-image of a Th-axiom (14), which represents an event of the form **"Go$_\alpha$"**.

By induction, starting with the root of D', we can prove that each of the non-terminal sequents in D' is of the form (here S stands for a 'timeless' part):

$$\left(\mathrm{T}(\tau)\otimes(\tau=h(\rho_0,\rho_1,\ldots,\rho_{\ell-1}))\otimes S\otimes\bigotimes_{i=0}^{\ell-1}(\rho_i\in\mathcal{D}_i)\otimes\bigotimes_\alpha d_\alpha(h_\alpha(\rho_0,\rho_1,\ldots,\rho_{\ell-1}))\right),\ \Gamma,\ !\Delta\vdash Z' \tag{35}$$

where $\rho_0,\rho_1,\ldots,\rho_{\ell-1}$ are distinct time variables, and each of these ρ_i is bound below by some rule $\exists$-**H**, which deals with an instance of the LL-image of the corresponding Th-axiom (14):

$$\dfrac{(Y_i\otimes(\rho_i\in\mathcal{D}'_\alpha)\otimes d_\alpha(\tau_i+\rho_i)\otimes V'),\ \Gamma',\ !\Delta'\vdash Z'}{(X_i\otimes d_\alpha(\infty)\otimes V'),\ ((X_i\otimes d_\alpha(\infty))\multimap(Y_i\otimes\exists\rho\,((\rho\in\mathcal{D}'_\alpha)\otimes d_\alpha(\tau_i+\rho)))),\ \Gamma',\ !\Delta'\vdash Z'}\ \exists\text{-}\mathbf{H} \tag{36}$$

here X_i is of the form $X_i = (\mathrm{T}(\tau_i)\otimes s)$, and Y_i is of the form $Y_i = (\mathrm{T}(\tau_i)\otimes\widetilde{s})$.

Table 1
The E-Horn Linear Logic. Both $\otimes$ and $\oplus$ are assumed to be commutative and associative.

$$\textbf{I}\quad \frac{}{(X\otimes V)\vdash X} \qquad\qquad \textbf{M}\quad \frac{X,\,\Gamma,\,!\Delta\vdash Y}{(X\otimes V),\,\Gamma,\,!\Delta\vdash (Y\otimes V)}$$

$$\textbf{E}\quad \frac{}{(X'\otimes V\otimes Z_j\otimes U(h))\vdash \exists t'(X'\otimes \overline{Z}^{\oplus}\otimes U(t'))}$$

where t' is a variable, h is a term, and $\overline{Z}^{\oplus}=(Z_1\oplus Z_2\oplus\cdots\oplus Z_k)$, and $1\le j\le k$.

$$\textbf{H}\quad \frac{(Y\otimes V),\,\Gamma,\,!\Delta\vdash Z'}{(X\otimes V),\,\Gamma,\,(X\multimap Y),\,!\Delta\vdash Z'}$$

$$\oplus\textbf{-H}\quad \frac{(Y_1\otimes V),\,\Gamma,\,!\Delta\vdash Z'\quad (Y_2\otimes V),\,\Gamma,\,!\Delta\vdash Z'\quad\ldots\quad (Y_m\otimes V),\,\Gamma,\,!\Delta\vdash Z'}{(X\otimes V),\,\Gamma,\,(X\multimap (Y_1\oplus Y_2\oplus\cdots\oplus Y_m)),\,!\Delta\vdash Z'}$$

$$\exists\textbf{-H}\quad \frac{(Y\otimes U(\rho)\otimes V),\,\Gamma,\,!\Delta\vdash Z'}{(X\otimes V),\,\Gamma,\,(X\multimap (Y\otimes \exists\rho\, U(\rho))),\,!\Delta\vdash Z'}$$

where ρ is a variable having no free occurrences in Γ, Δ, Y, V, X, and Z'.

$$\forall\textbf{-H}\quad \frac{V,\,\Gamma,\,(X(h)\multimap Y(h)),\,!\Delta\vdash Z'}{V,\,\Gamma,\,\forall z\,(X(z)\multimap Y(z)),\,!\Delta\vdash Z'}$$

$$\textbf{L!}\quad \frac{X,\,\Gamma,\,A,\,!\Delta\vdash Z'}{X,\,\Gamma,\,!A,\,!\Delta\vdash Z'}\qquad \textbf{W!}\quad \frac{X,\,\Gamma,\,!\Delta\vdash Z'}{X,\,\Gamma,\,!A,\,!\Delta\vdash Z'}\qquad \textbf{C!}\quad \frac{X,\,\Gamma,\,!A,\,!A,\,!\Delta\vdash Z'}{X,\,\Gamma,\,!A,\,!\Delta\vdash Z'}$$

$$\textbf{Cut}\quad \frac{X,\,\Gamma_1,\,!\Delta_1\vdash U\qquad U,\,\Gamma_2,\,!\Delta_2\vdash Z'}{X,\,\Gamma_1,\,\Gamma_2,\,!\Delta_1,\,!\Delta_2\vdash Z'}$$

Furthermore, by the appropriate commuting conversions we can push an ($\oplus$**-H**)-rule of the form:

$$\frac{\dfrac{\pi_1}{((\rho_i\in\mathcal{E}_1)\otimes V),\,\Gamma,\,!\Delta\vdash Z'}\qquad \dfrac{\pi_2}{((\rho_i\in\mathcal{E}_2)\otimes V),\,\Gamma,\,!\Delta\vdash Z'}}{((\rho_i\in\mathcal{E})\otimes V),\,((\rho_i\in\mathcal{E})\multimap((\rho_i\in\mathcal{E}_1)\oplus(\rho_i\in\mathcal{E}_2))),\,\Gamma,\,!\Delta\vdash Z'}\,\oplus\textbf{-H}$$

downwards to the corresponding rule (36) that binds the ρ_i.
Notice that a combination of consecutive ($\oplus$**-H**)-rules with the same ρ_i, such as

$$\frac{\dfrac{\dfrac{\pi_1}{((\rho_i\in\mathcal{E}_1)\otimes V),\,\Gamma,\,!\Delta\vdash Z'}\qquad \dfrac{\pi_2}{((\rho_i\in\mathcal{E}_2)\otimes V),\,\Gamma,\,!\Delta\vdash Z'}}{((\rho_i\in\mathcal{E}')\otimes V),\,A',\,\Gamma,\,!\Delta\vdash Z'}\,\oplus\textbf{-H}\qquad \dfrac{\pi''}{((\rho_i\in\mathcal{E}'')\otimes V),\,A',\,\Gamma,\,!\Delta\vdash Z'}}{((\rho_i\in\mathcal{E})\otimes V),\,A,\,A',\,\Gamma,\,!\Delta\vdash Z'}\,\oplus\textbf{-H}$$

where

$$A=((\rho_i\in\mathcal{E})\multimap((\rho_i\in\mathcal{E}')\oplus(\rho_i\in\mathcal{E}'')))$$

and

$$A'=((\rho_i\in\mathcal{E}')\multimap((\rho_i\in\mathcal{E}_1)\oplus(\rho_i\in\mathcal{E}_2))),$$

can be glued into one ($\oplus$**-H**)-rule with the same ρ_i:

$$\frac{\dfrac{\pi_1}{((\rho_i\in\mathcal{E}_1)\otimes V),\,\Gamma,\,!\Delta\vdash Z'}\qquad \dfrac{\pi_2}{((\rho_i\in\mathcal{E}_2)\otimes V),\,\Gamma,\,!\Delta\vdash Z'}\qquad \dfrac{\pi''}{((\rho_i\in\mathcal{E}'')\otimes V),\,\Gamma,\,!\Delta\vdash Z'}}{((\rho_i\in\mathcal{E})\otimes V),\,\widetilde{A},\,\Gamma,\,!\Delta\vdash Z'}\,\oplus\textbf{-H}$$

with

$$\widetilde{A}=((\rho_i\in\mathcal{E})\multimap((\rho_i\in\mathcal{E}_1)\oplus(\rho_i\in\mathcal{E}_2)\oplus(\rho_i\in\mathcal{E}''))).$$

By repeatedly applying commuting conversions and gluing, we come down to (36), resulting in something like this (for brevity, $\widetilde{V}_i$ stands for $(Y_i\otimes d_\alpha(\tau_i+\rho_i)\otimes V')$):

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi_1}{((\rho_i\in\mathcal{E}_1)\otimes\widetilde{V}_i),\,\Gamma',\,!\Delta'\vdash Z'}\qquad \dfrac{\pi_2}{((\rho_i\in\mathcal{E}_2)\otimes\widetilde{V}_i),\,\Gamma',\,!\Delta'\vdash Z'}\qquad \dfrac{\pi_3}{((\rho_i\in\mathcal{E}_3)\otimes\widetilde{V}_i),\,\Gamma',\,!\Delta'\vdash Z'}}{((\rho_i\in\mathcal{D}'_\alpha)\otimes\widetilde{V}_i),\,((\rho_i\in\mathcal{D}'_\alpha)\multimap((\rho_i\in\mathcal{E}_1)\oplus(\rho_i\in\mathcal{E}_2)\oplus(\rho_i\in\mathcal{E}_3))),\,\Gamma',\,!\Delta'\vdash Z'}\,\oplus\textbf{-H}}{((\rho_i\in\mathcal{D}'_\alpha)\otimes\widetilde{V}_i),\,\forall\rho\,((\rho\in\mathcal{D}'_\alpha)\multimap((\rho\in\mathcal{E}_1)\oplus(\rho\in\mathcal{E}_2)\oplus(\rho\in\mathcal{E}_3))),\,\Gamma',\,!\Delta'\vdash Z'}\,\forall\textbf{-H}}{((\rho_i\in\mathcal{D}'_\alpha)\otimes\widetilde{V}_i),\,!\forall\rho\,((\rho\in\mathcal{D}'_\alpha)\multimap((\rho\in\mathcal{E}_1)\oplus(\rho\in\mathcal{E}_2)\oplus(\rho\in\mathcal{E}_3))),\,\Gamma',\,!\Delta'\vdash Z'}\,\textbf{L!}}{((\rho_i\in\mathcal{D}'_\alpha)\otimes\widetilde{V}_i),\,\Gamma',\,!\Delta'\vdash Z'}\,\textbf{C!}}{(X_i\otimes d_\alpha(\infty)\otimes V'),\,((X_i\otimes d_\alpha(\infty))\multimap(Y_i\otimes\exists\rho\,((\rho\in\mathcal{D}'_\alpha)\otimes d_\alpha(\tau_i+\rho)))),\,\Gamma',\,!\Delta'\vdash Z'}\,\exists\textbf{-H}$$

where $\mathcal{D}'_\alpha\subseteq\mathcal{E}_1\cup\mathcal{E}_2\cup\mathcal{E}_3$, and no ($\oplus$**-H**)-rule with this ρ_i is applied inside π_1, π_2, and π_3.

Suppose that $\mathcal{W}_1$, $\mathcal{W}_2$, $\mathcal{W}_3$ are winning strategies that have been already associated with π_1, π_2, π_3, respectively.

With this vertex (36) in D', which is of the form:

$$\frac{((\rho_i \in \mathcal{D}'_\alpha) \otimes \widetilde{V}_i),\ \Gamma',\ !\Delta' \vdash Z'}{(X_i \otimes d_\alpha(\infty) \otimes V'),\ ((X_i \otimes d_\alpha(\infty)) \multimap (Y_i \otimes \exists \rho\, ((\rho \in \mathcal{D}'_\alpha) \otimes d_\alpha(\tau_i + \rho)))),\ \Gamma',\ !\Delta' \vdash Z'}\ \exists\text{-}\mathbf{H}$$

we associate then a winning strategy $\mathcal{W}_{(36)}$ of the form:

$$\begin{tikzcd}
 & (S, \mathrm{go}_\alpha, \tau_i) \arrow[dl, "\underline{(\rho_i \in \mathcal{E}_1)}"'] \arrow[d, "\underline{(\rho_i \in \mathcal{E}_2)}"] \arrow[dr, "\underline{(\rho_i \in \mathcal{E}_3)}"] & \\
\mathcal{W}_1 & \mathcal{W}_2 & \mathcal{W}_3
\end{tikzcd}$$

The remaining cases are treated in a similar way. $\square$

Remark 12.1. Theorem 12.1 is dealing with certain sets of time moments and functions from time moments into time moments. The theory Th includes certain 'time axioms' about such sets and their unions (see Section 2).

In fact, Theorem 12.1 provides an exact correlation between two levels:

(a) the level of sets of reals $\mathcal{D}$ and real functions involved in winning strategies (see Lemma 12.1), and

(b) the level of sets of reals $\mathcal{E}$ and real functions involved in the axioms of real time, for instance, as atomic formulas of the form $(\rho \in \mathcal{E})$.

From the practical point of view (see, for instance, [2,10,7,3,4,9]), the most interesting case is the case where we are dealing with Boolean combinations of time intervals and with Boolean combinations of linear functions.

Corollary 12.1. *Let* Th *be an affine logic theory that includes as its proper axioms the Horn-like specifications of all actions α, and the axioms of real time in the form of linear equalities/inequalities (see Section 2).*

Then a sequent of the form (29):

$$\left(\mathtt{T}(0) \otimes W \otimes \bigotimes_\alpha d_\alpha(\infty)\right) \vdash \exists t' \left((A_0 \le t' \le B_0) \otimes \mathtt{T}(t') \otimes \overline{Z}^{\oplus} \otimes \bigotimes_\alpha d_\alpha(\infty)\right),$$

is provable in Th *if and only if there exists a winning strategy $\mathcal{W}$ in a finite concise form, in which all sets of reals $\mathcal{D}$ involved are Boolean combinations of intervals, and all functions involved are piecewise linear functions.*

Moreover, there is a direct correspondence between Th*-proofs for this sequent and winning strategies of this kind.*

13. E-Horn linear logic derivations

In this section we introduce specific affine logic rules, the system of which will have the sufficient strength to handle the planning problems under temporal uncertainty (see Theorem 12.1):

Definition 13.1. Below X, X', Y, Y_i, U, V, Z_j stand for elementary products of atomic predicate formulas, the connectives $\otimes$ and $\oplus$ are assumed to be commutative and associative.

(a) `'axiom'`:

$$\overline{(X \otimes V) \vdash X}$$

(b) `'E-axiom'`:

$$\overline{(X' \otimes V \otimes Z_j \otimes U(h)) \vdash \exists t'(X' \otimes \overline{Z}^{\oplus} \otimes U(t'))},$$

where t' is a 'time variable', h is a 'time term', and $\overline{Z}^{\oplus} = (Z_1 \oplus Z_2 \oplus \cdots \oplus Z_k)$, and $1 \le j \le k$.

(c) `'ax-cut'`:

$$\frac{X \vdash Y \qquad (Y \otimes V) \vdash Z'}{(X \otimes V) \vdash Z'},$$

where $X \vdash Y$ is an axiom of a given theory Th, and Z' is either an elementary product of atomic predicate formulas, or a formula of the form $(X' \otimes \overline{Z}^{\oplus})$, or a formula of the form $\exists t'(X' \otimes \overline{Z}^{\oplus} \otimes U(t'))$.

(d) `'∃-ax-cut'`:

$$\frac{X \vdash (Y \otimes \exists \rho\, U(\rho)) \qquad (Y \otimes V \otimes U(\rho')) \vdash Z'}{(X \otimes V) \vdash Z'},$$

where $X \vdash (Y \otimes \exists \rho\, U(\rho))$ is an axiom of Th, and ρ' is a 'time variable' having no occurrence in Y, V, and Z'.

(e) '$\oplus$-ax-cut':

$$\frac{X \vdash (Y_1 \oplus Y_2 \oplus \cdots \oplus Y_m) \qquad (Y_1 \otimes V) \vdash Z' \quad (Y_2 \otimes V) \vdash Z' \quad \ldots \quad (Y_m \otimes V) \vdash Z'}{(X \otimes V) \vdash Z'},$$

where $X \vdash (Y_1 \oplus Y_2 \oplus \cdots \oplus Y_m)$ is an axiom of Th.

(f) 'cut':

$$\frac{X \vdash Y \qquad (Y \otimes V) \vdash Z'}{(X \otimes V) \vdash Z'}$$

(Notice that the 'cut formula' Y is confined to an elementary product of atomic predicate formulas.) □

The above system in Definition 13.1 turns out to be complete with respect to Th-provability for 'task sequents' of the form (29). To simplify technicalities, first, we translate Th-proofs into the corresponding proofs within pure affine logic.

Definition 13.2. Given a system with a finite set of actions with delayed effects, let Th be an affine logic theory that includes as its proper axioms the Horn-like specifications of all actions α, and the appropriate axioms of real time (see Section 2). Each of the non-logical axioms of Th of the form

$$X(z_1, \ldots, z_n) \vdash Y(z_1, \ldots, z_n) \tag{37}$$

will be encoded as its *'LL-image'*:

$$\forall z_1 \cdots z_n \,(X(z_1, \ldots, z_n) \multimap Y(z_1, \ldots, z_n)). \tag{38}$$

Proposition 13.1. *Any* Th*-proof* D_{Th} *for a sequent of the form:*

$$\Gamma \vdash C$$

can be easily transformed into a purely affine logic proof for the following sequent:

$$\Gamma,\ !\Delta \vdash C$$

where Δ *consists of the LL-images of all non-logical axioms of* Th *that participate in* D_{Th}*, and vice versa.*

Then we will use a purely linear logic E-Horn version represented in Table 1 to provide the Horn-like completeness, which we need to complete the part: **"Proofs $\Longrightarrow$ Strategies,"** in Theorem 12.1.

Lemma 13.1. *Let* D_{Th} *be a* Th*-proof for a 'task sequent' of the form (recall that each action* α *is supplied with the 'time-guarded' predicate* $d_\alpha(x)$*)*

$$\left(\mathtt{T}(0) \otimes W \otimes \bigotimes_{\alpha} d_\alpha(\infty)\right) \vdash \exists t' \left((A_0 \le t' \le B_0) \otimes \mathtt{T}(t') \otimes \overline{Z}^{\oplus} \otimes \bigotimes_{\alpha} d_\alpha(\infty)\right) \tag{39}$$

where $\overline{Z}^{\oplus} = (Z_1 \oplus Z_2 \oplus \cdots \oplus Z_k)$.

Let Δ *consist of the LL-images of all non-logical axioms of* Th *that participate in* D_{Th}.

Assume this Th*-proof* D_{Th} *be translated into a purely affine logic proof* D_{AL} *for a sequent of the form:*

$$\left(\mathtt{T}(0) \otimes W \otimes \bigotimes_{\alpha} d_\alpha(\infty)\right),\ !\Delta \vdash \exists t' \left((A_0 \le t' \le B_0) \otimes \mathtt{T}(t') \otimes \overline{Z}^{\oplus} \otimes \bigotimes_{\alpha} d_\alpha(\infty)\right). \tag{40}$$

Notice that the planning task expressed by (40) is the same planning task expressed by (39).

Then such a D_{AL} *can be rearranged to apply only the Horn-like rules taken from Table 1, where* X, X', Y, Y_i, U, V, Z_j *stand for elementary products of atomic predicate formulas,* Z' *is either an elementary product of atomic predicate formulas, or a formula of the form* $\exists t'(X' \otimes \overline{Z}^{\oplus} \otimes U(t'))$*, and* Γ *consists of the LL-images (and their instances) of non-logical axioms of* Th*:*

$$\forall z_1 \cdots z_n \,(X(z_1, \ldots, z_n) \multimap Y(z_1, \ldots, z_n)),$$

and/or their instances with some terms $h_1, \ldots, h_n$*:*

$$(X(h_1, \ldots, h_n) \multimap Y(h_1, \ldots, h_n)).$$

Proof. Any Th-proof for (39) can be easily translated into a cut-free purely affine logic proof D' for (40), and vice versa. The cut-free D' can use only the following rules from Table 2 (we cluster them in two groups):

(i) "Left rules": **L**$\otimes$, **L**$\oplus$, **L**$\multimap$, **L!**, **W!**, **C!**, **L**$\exists$, **L**$\forall$.
(ii) "Right rules": **R**$\otimes$, **R**$\oplus$, **R**$\exists$.

Now we will transform the cut-free proof D' whose rules are from Table 2 into a proof in which the rules are taken from the Horn-like Table 1.

First, we push our "right rules" upwards (to logical axioms) and push our "left rules" downwards (to the conclusion) by repeatedly applying the corresponding *commuting conversions*:

(a) E.g., a combination: "first **L**$\oplus$, then **R**$\otimes$," of the form (here π_0, π_1 and π_2 are proofs):

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma', Y_1 \vdash Z'} \quad \dfrac{\pi_2}{\Gamma', Y_2 \vdash Z'}}{\Gamma', (Y_1 \oplus Y_2) \vdash Z'}\,\mathbf{L}\oplus \qquad \dfrac{\pi_0}{\Gamma'' \vdash Z''}}{\Gamma', (Y_1 \oplus Y_2), \Gamma'' \vdash (Z' \otimes Z'')}\,\mathbf{R}\otimes$$

can be replaced with the following combination: "first **R**$\otimes$, then **L**$\oplus$:"

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma', Y_1 \vdash Z'} \quad \dfrac{\pi_0}{\Gamma'' \vdash Z''}}{\Gamma', Y_1, \Gamma'' \vdash (Z' \otimes Z'')}\,\mathbf{R}\otimes \qquad \dfrac{\dfrac{\pi_2}{\Gamma', Y_2 \vdash Z'} \quad \dfrac{\pi_0}{\Gamma'' \vdash Z''}}{\Gamma', Y_2, \Gamma'' \vdash (Z' \otimes Z'')}\,\mathbf{R}\otimes}{\Gamma', (Y_1 \oplus Y_2), \Gamma'' \vdash (Z' \otimes Z'')}\,\mathbf{L}\oplus$$

(b) A combination: "first **L**$\multimap$, then **R**$\otimes$," of the form (here π_0, π_1 and π_2 are proofs):

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma_1 \vdash X} \quad \dfrac{\pi_2}{Y, \Gamma_2 \vdash Z'}}{\Gamma_1, (X \multimap Y), \Gamma_2 \vdash Z'}\,\mathbf{L}\multimap \qquad \dfrac{\pi_0}{\Gamma'' \vdash Z''}}{\Gamma_1, (X \multimap Y), \Gamma_2, \Gamma'' \vdash (Z' \otimes Z'')}\,\mathbf{R}\otimes$$

can be replaced with the following combination: "first **R**$\otimes$, then **L**$\multimap$:"

$$\dfrac{\dfrac{\pi_1}{\Gamma_1 \vdash X} \qquad \dfrac{\dfrac{\pi_2}{Y, \Gamma_2 \vdash Z'} \quad \dfrac{\pi_0}{\Gamma'' \vdash Z''}}{Y, \Gamma_2, \Gamma'' \vdash (Z' \otimes Z'')}\,\mathbf{R}\otimes}{\Gamma_1, (X \multimap Y), \Gamma_2, \Gamma'' \vdash (Z' \otimes Z'')}\,\mathbf{L}\multimap$$

(c) The remaining combinations are treated in a similar way.

As a result, the "right rules" will be applied only to axioms, which allows to establish the explicit form of the corresponding sequents.

In particular, suppose the **R**$\exists$-rule is applied as follows (here $\overline{Z}^{\oplus} = (Z_1 \oplus Z_2 \oplus \cdots \oplus Z_k)$, and π is a proof in which no "left rule" is applied):

$$\dfrac{\dfrac{\pi}{\Phi \vdash (X' \otimes \overline{Z}^{\oplus} \otimes U(h))}\,\mathbf{R}\ldots}{\Phi \vdash \exists t'(X' \otimes \overline{Z}^{\oplus} \otimes U(t'))}\,\mathbf{R}\exists \tag{41}$$

Then Φ must be of the form (modulo associativity and commutativity of $\otimes$):

$$\Phi = (X' \otimes V \otimes Z_j \otimes U(h)). \tag{42}$$

Hence, by pruning away parts such as $\dfrac{\pi}{\Phi \vdash (X' \otimes \overline{Z}^{\oplus} \otimes U(h))}$ in (41), we obtain the proof in which the leaves are sequents of the form **E** or **I** declared as axioms in Table 1.

As for the "left rules", by induction we can simulate each of these rules with the Horn-like rules from Table 1:

(a) E.g., an (**L**$\multimap$)-rule of the form (here π_1 and π_2 are proofs that have been already constructed by induction with rules from Table 1):

$$\dfrac{\dfrac{\pi_1}{X', \Gamma_1, \,!\Delta_1 \vdash X} \quad \dfrac{\pi_2}{Y, V, \Gamma_2, \,!\Delta_2 \vdash Z'}}{X', \Gamma_1, \,!\Delta_1, (X \multimap Y), V, \Gamma_2, \,!\Delta_2 \vdash Z'}\,\mathbf{L}\multimap$$

is simulated with the following Horn-like rules from Table 1:

$$\dfrac{\dfrac{\dfrac{\pi_1}{X', \Gamma_1, \,!\Delta_1 \vdash X}}{(X' \otimes V), \Gamma_1, \,!\Delta_1 \vdash (X \otimes V)}\,\mathbf{M} \qquad \dfrac{\dfrac{\pi_2}{(Y \otimes V), \Gamma_2, \,!\Delta_2 \vdash Z'}}{(X \otimes V), \Gamma_2, (X \multimap Y), \,!\Delta_2 \vdash Z'}\,\mathbf{H}}{(X' \otimes V), \Gamma_1, \Gamma_2, (X \multimap Y), \,!\Delta_1, \,!\Delta_2 \vdash Z'}\,\mathbf{Cut}$$

(b) As for the (**L**$\oplus$)-rule, by the appropriate commuting conversions we can push it downwards (to the related **L**$\multimap$), resulting in something like this:

$$\dfrac{\dfrac{\pi_0}{X', \Gamma', \,!\Delta' \vdash X} \qquad \dfrac{\dfrac{\pi_1}{Y_1, V, \Gamma, \,!\Delta \vdash Z'} \dfrac{\pi_2}{Y_2, V, \Gamma, \,!\Delta \vdash Z'}}{(Y_1 \oplus Y_2), V, \Gamma, \,!\Delta \vdash Z'}\,\mathbf{L}\oplus}{X', \Gamma', \,!\Delta', (X \multimap (Y_1 \oplus Y_2)), V, \Gamma, \,!\Delta \vdash Z'}\,\mathbf{L}\multimap$$

which is simulated with the following Horn-like rules from Table 1:

$$\dfrac{\dfrac{\dfrac{\pi_0}{X', \Gamma', \,!\Delta' \vdash X}}{(X' \otimes V), \Gamma', \,!\Delta' \vdash (X \otimes V)}\,\mathbf{M} \qquad \dfrac{\dfrac{\pi_1}{(Y_1 \otimes V), \Gamma, \,!\Delta \vdash Z'} \quad \dfrac{\pi_2}{(Y_2 \otimes V), \Gamma, \,!\Delta \vdash Z'}}{(X \otimes V), \Gamma, (X \multimap (Y_1 \oplus Y_2)), \,!\Delta \vdash Z'}\,\oplus\text{-}\mathbf{H}}{(X' \otimes V), \Gamma', \Gamma, (X \multimap (Y_1 \oplus Y_2)), \,!\Delta', \,!\Delta \vdash Z'}\,\mathbf{Cut}$$

(c) Similarly, the ($\mathbf{L}\,\exists$)-rule can be pushed downwards (to the related $\mathbf{L}\!\multimap$), resulting in something like this:

$$\dfrac{\dfrac{\pi_0}{X', \Gamma', \,!\Delta' \vdash X} \qquad \dfrac{\dfrac{\pi_1}{Y, U(\rho), V, \Gamma, \,!\Delta \vdash Z'}}{Y, \exists\rho\, U(\rho), V, \Gamma, \,!\Delta \vdash Z'}\,\mathbf{L}\,\exists}{X', \Gamma', \,!\Delta', (X \multimap (Y \otimes \exists\rho\, U(\rho))), V, \Gamma, \,!\Delta \vdash Z'}\,\mathbf{L}\!\multimap$$

which is simulated with the following Horn-like rules from Table 1:

$$\dfrac{\dfrac{\dfrac{\pi_0}{X', \Gamma', \,!\Delta' \vdash X}}{(X' \otimes V), \Gamma', \,!\Delta' \vdash (X \otimes V)}\,\mathbf{M} \qquad \dfrac{\dfrac{\pi_1}{(Y \otimes V), U(\rho), \Gamma, \,!\Delta \vdash Z'}}{(X \otimes V), \Gamma, (X \multimap (Y \otimes \exists\rho\, U(\rho))), \,!\Delta \vdash Z'}\,\exists\text{-}\mathbf{H}}{(X' \otimes V), \Gamma', \Gamma, (X \multimap (Y \otimes \exists\rho\, U(\rho))), \,!\Delta', \,!\Delta \vdash Z'}\,\mathbf{Cut}$$

(d) The remaining cases are treated in a straightforward way. □

14. Concluding remarks

We have introduced the E-Horn fragment of linear logic (see Section 13) as a comprehensive logical system capable of handling the typical AI problem of making a plan of the actions to be performed by a controller so that it could get into a set of final situations, if it started with a certain initial situation.

A particular focus of this paper is on planning problems in which actions may have *quantitatively delayed effects*, and where the delays are non-deterministic and continuous.

We have shown that the potentially unbounded winning strategies which may arise in this context can be exactly captured by proofs within the E-Horn fragment of linear logic. Within this paradigm "proofs $\Longleftrightarrow$ plans," we have established thereby a comprehensive and adequate logical model of strong planning under temporal uncertainty which addresses infinity concerns.

One could say that some examples given in the paper (e.g., Example 1.1) could be reformulated in several planning formalisms, such as timed transition systems, timed automata, Markov decision processes, etc (see, for instance, [1]).

What arguments could we offer about the superiority of linear logic as a modeling formalism for the planning domains under temporal uncertainty considered here (save its being a novel compared to the others) ?

(A) The simple Horn fragment of linear logic provides ordinary users with an easy way of specifying their robot systems in their own terms, without radical reformulation of the original problem. In particular, our choice of the minimal set of connectives: $\otimes$ and $\oplus$, is in full accordance with the naive AI semantics.

(B) By means of the linear logic proof machinery we have overcome the basic obstruction to finding an adequate logical formalism for the planning under temporal uncertainty: the discrepancy between *global* quantitative time constraints and the *locality*, or *memoryless*, property of the proofs - that any inference rule depends *only* on its direct premises.

(C) As for the timed automata [1], they are very efficient for the so-called 'reachability properties' such as safety property: *"there is no path to a given state S."*

For the actions with non-deterministic delays, we are involved in a certain *game* against Nature: In order to succeed, we have to respond properly to *each* of the infinite number of possible delays on the road from the initial state to a final one.

Hence, to make a timed automata approach appropriate, we have to provide an infinite number of links between some nodes of a timed automaton. On top of that, we have to distinguish certain links as controlled by a robot and 'delay' links as 'controlled' by Nature.

In other words, we need the concept of an *alternating timed automaton A* and a winning strategy on *A*. For the sake of brevity, we confine ourselves to a tree-based formalism for winning strategies (see Definition 4.1).

(D) As for the Markov decision processes, we cannot directly apply their techniques, since the basic point of their theory - that is, given the state at time t is known, transition probabilities to the state at time $t+1$ are *independent* of all previous states or actions. Recall that the winning strategies under consideration may use the *whole* pre-history information on the branch leading from the root to a given state.

On the other hand, it is promising to take into account the distribution of possible delays to provide both non-deterministic and probabilistic approaches.

(E) Linear logic is capable of coping with *preemptive/anticipative planning*, as in Example 1.1, in a very natural way.

In order to capture such *preemptive/anticipative plans* within timed automata or Markov decision processes, we have to consider all possible timed combinations of partial states, which leads to a combinatorial explosion in number of states even on the level of specifications.

As for the complexity of proof-search in the proposed fragment of affine logic, this is not an object of this paper (our idea is to find an exact logical model for the time planning). Our intention is to address the complexity issues in the next paper, as it has been done for the actions with instant effects[15,16].

Theorem 12.1 remains valid even for planning problems in which actions may have *non-deterministic effects* with *quantitatively delayed effects*, and where the delays are non-deterministic and continuous.

Appendix. Linear logic rules

In fact, we can confine ourselves to the intuitionistic version of the first-order linear logic enriched with the Weakening rule **W** (see Table 2).

Table 2
The inference rules of intuitionistic affine logic.

I	$\dfrac{}{A \vdash A}$	**P**	$\dfrac{\Gamma,\ A, B,\ \Delta \vdash C}{\Gamma,\ B, A,\ \Delta \vdash C}$
Cut	$\dfrac{\Gamma \vdash A \quad A, \Delta \vdash C}{\Gamma, \Delta \vdash C}$	**W**	$\dfrac{\Gamma \vdash B}{\Gamma, A \vdash B}$
L⊗	$\dfrac{\Gamma, A, B \vdash C}{\Gamma, (A \otimes B) \vdash C}$	**R⊗**	$\dfrac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash (A \otimes B)}$
L⊕	$\dfrac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, (A \oplus B) \vdash C}$	**R⊕**	$\dfrac{\Gamma \vdash A}{\Gamma \vdash (A \oplus B)} \quad \dfrac{\Gamma \vdash B}{\Gamma \vdash (A \oplus B)}$
L⊸	$\dfrac{\Gamma \vdash A \quad B, \Delta \vdash C}{\Gamma, (A \multimap B), \Delta \vdash C}$	**R⊸**	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash (A \multimap B)}$
L1	$\dfrac{\Gamma \vdash C}{\Gamma, 1 \vdash C}$	**R1**	$\dfrac{}{\vdash 1}$
L! $\dfrac{\Gamma, A \vdash C}{\Gamma,\ !A \vdash C}$ **W!** $\dfrac{\Gamma \vdash C}{\Gamma,\ !A \vdash C}$	**C!** $\dfrac{\Gamma,\ !A,\ !A \vdash C}{\Gamma,\ !A \vdash C}$	**R!**	$\dfrac{!\Delta \vdash C}{!\Delta \vdash\ !C}$
L∃	$\dfrac{\Gamma, A(v) \vdash C}{\Gamma, \exists v\, A(v) \vdash C}$ (v is not free in Γ and C)	**R∃**	$\dfrac{\Gamma \vdash C(s)}{\Gamma \vdash \exists v\, C(v)}$ (s is a term)
L∀	$\dfrac{\Gamma, A(s) \vdash C}{\Gamma, \forall v\, A(v) \vdash C}$ (s is a term)	**R∀**	$\dfrac{\Gamma \vdash C(v)}{\Gamma \vdash \forall v\, C(v)}$ (v is not free in Γ)

References

[1] R. Alur, D.L. Dill, A theory of timed automata, Theoretical Computer Science 126 (1994) 183–235.
[2] C. Boutilier, T. Dean, S. Koenig, Editorial, Artificial Intelligence 147 (2003) 1–4.
[3] A. Cimatti, M. Pistore, M. Roveri, P. Traverso, Weak, strong, and strong cyclic planning via symbolic model checking, Artificial Intelligence 147 (2003) 35–84.
[4] K. Erol, D. Nau, J. Hendler, A critical look at critics in HTN planning, in: IJCAI-95, Montreal, August, 1995.
[5] A. Coles, M. Fox, K. Halsey, D. Long, A. Smith, Managing Concurrency in Temporal Planning using planner-scheduler interaction, Artificial Intelligence 173 (2009) 1–44.
[6] W. Cushing, S. Kambhampati, Mausam, D. Weld, When is temporal planning really temporal planning? In IJCAI-07, Hyderabad, A.P., India. January 2007, pp. 1852–1859, 2007.
[7] D. McDermott, J. Hendler, Planning: what it is, what it could be, an introduction to the special issue on planning and scheduling, Artificial Intelligence 76 (1995) 1–16.
[8] A. Garrido, M. Fox, D. Long, A temporal planning system for durative actions of PDDL2.1, ECAI 2002, pp. 586–590.
[9] H. Geffner, Functional Strips: a more flexible language for planning and problem solving, in: Logic-based Artificial Intelligence, in: Kluwer International Series In Engineering and Computer Science, 2000, pp. 187–209.
[10] M. Ghallab, D. Nau, P. Traverso, Automated Planning: Theory and Practice, Morgann Kaufmann, 2004.
[11] J.-Y. Girard, Linear logic, Theoretical Computer Science 50 (1) (1987) 1–102.
[12] J.-Y. Girard, Linear logic: its syntax and semantics, in: J.-Y. Girard, Y. Lafont, L. Regnier (Eds.), Advances in Linear Logic, in: London Mathematical Society Lecture Notes, vol. 222, Cambridge University Press, 1995, pp. 1–42.
[13] M.I. Kanovich, Horn Programming in Linear Logic is NP-complete, in: Proc. 7th Annual IEEE Symposium on Logic in Computer Science, Santa Cruz, June 1992, pp. 200–210.
[14] Max Kanovich, Linear logic as a logic of computations, Annals of Pure and Applied Logic 67 (1994) 183–212.
[15] Max Kanovich, Jacqueline Vauzeilles, The Classical AI Planning Problems in the Mirror of Horn Linear Logic: Semantics, Expressibility, Complexity, in: Math. Struct. in Comp. Science, vol. 11, 2001, pp. 689–716.
[16] Max Kanovich, Jacqueline Vauzeilles, Strong planning under uncertainty in domains with numerous but identical elements (a generic approach), Theoretical Computer Science 379 (2007) 84–119.

[17] H. Kautz, B. Selman, Pushing the envelope: planning, propositional logic, and stochastic search, in: Proc. AAAI-1996, Portland, OR.
[18] D. Long, M. Fox, Exploiting a graphplan framework in temporal planning, in: ICAPS 2003, pp. 52–61.
[19] M. Masseron, C. Tollu, J. Vauzeilles, Generating plans in linear logic: I. Actions as proofs, Theoretical Computer Science 113 (1993) 349–370.
[20] M. Masseron, Generating plans in linear logic: II. A geometry of action, Theoretical Computer Science 113 (1993) 371–375.
[21] E. Noether, Invariante Varlationsprobleme, Nachr. d. Knig. Gesellsch. d. Wiss. zu Gttingen, Math-phys. Klasse, 1918, pp. 235-257; English translation M. A. Travel, Transport Theory and Statistical Physics 1 (3) (1971) 183–207.
[22] N.J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, 1998.