Math. Struct. in Comp. Science (2001), *vol.* 11, *pp.* 689–716. © 2001 Cambridge University Press
DOI: 10.1017/S0960129501003413 Printed in the United Kingdom

The classical AI planning problems in the mirror of Horn linear logic: semantics, expressibility, complexity

MAX KANOVICH[†] and JACQUELINE VAUZEILLES[‡]

[†] *Russian State University for the Humanities,*
Miusskaya 6, 125267 Moscow, Russia.
Email: max@kanovich.dnttm.rssi.ru, mk@lipn.univ-paris13.fr

[‡] *LIPN, UPRESA CNRS 7030, Institut Galilée,*
99 Avenue J.-B.Clément, 93430 Villetaneuse, France.
Email: jv@lipn.univ-paris13.fr

Received 15 December 1999; revised 20 September 2000

We introduce Horn linear logic as a comprehensive logical system capable of handling the typical AI problem of making a plan of the actions to be performed by a robot so that he could get into a set of final situations, if he started with a certain initial situation.
Contrary to undecidability of propositional Horn linear logic, the planning problem is proved to be decidable for a reasonably wide class of natural robot systems.
The planning problem is proved to be EXPTIME-complete for the robot systems that allow actions with non-deterministic effects. Fixing a finite signature, that is a finite set of predicates and their finite domains, we get a *polynomial time* procedure of making plans for the robot system over this signature.
The planning complexity is reduced to PSPACE for the robot systems with only pure deterministic actions.
As *honest* numerical parameters in our algorithms we invoke the length of description of a planning task 'from W to $\widetilde{Z}$' and the Kolmogorov descriptive complexity of $\mathtt{Ax}_T$, a set of possible actions.

1. Introduction and Summary

The potential usefulness of a logical approach to AI problems (Nilsson 1980; Bibel 1986; McCarthy 1986) has motivated attempts to achieve higher efficiency than that provided by traditional predicate logic. It is true that one could describe any AI system by means of formulas built up from certain basic predicates using traditional logical connectives and quantifiers. However, such a general approach runs into difficulties with the undecidability of certain fragments of traditional predicate logic, and it is not easy to describe a decidable

[†] Initial results obtained during a visit to Université de Paris-Nord in Summer 1998.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

fragment of it that could be used for specifying AI systems and their problems (*cf*., for instance, the connection method in Bibel (1986)).

The aim of this research is to introduce a comprehensive logical system that automatically exploits peculiarities of the AI systems under consideration, and to show that such a logical system can indeed achieve a significant speedup over the traditional ones.

The AI systems we consider have the following features.

A *robot* is dealing with a finite number of *objects*. Each of the *actions* performed by the robot results in correlations *newly established* between the objects, with some old correlations being *destroyed*. The conditions *enabling* an action consist of the presence of certain old correlations.

The typical AI problem is that of *making a plan* of the actions to be performed by the robot so that it could get into a set of *final situations* if it started with a certain *initial situation*(see Nilsson (1980), Bibel (1986), Chapman (1987), Masseron *et al.* (1993), and McDermott and Hendler (1995)).

Extending Nilsson (1980), Bibel (1986), Chapman (1987), and McDermott and Hendler (1995), we consider here *two kinds of non-determinism* related to the points from which the robot activity could be regarded (Masseron *et al.* 1993).

(1) The robot makes its *own choice* between the actions from a given set.

(2) The effect of the action chosen may be *non-deterministic* because the robot does not know, *in advance*, what the reaction of Nature would be under the particular circumstances.

Indeed, in the latter case the corresponding plans are conceived of as *winning strategies* against Nature.

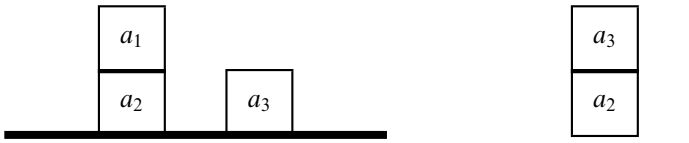
Let us start with a classical example from Nilsson (1980).

Example 1.1. (**Blocks World**) A one-handed robot deals with a number of cubic blocks $a_1, a_2, \ldots, a_N$.
An *initial* situation is that a_2 and a_3 are on a table, and a_1 is on the top of a_2.
Actions the robot might take are as follows:

(i) `stack`(x, y): 'holding block x in its hand, put x down on the clear top of block y, and leave the hand empty',
(ii) `unstack`(x, y): 'having the clear block x stacked on y, pick x up by the empty hand, and hold x in the hand',
(iii) `pickup`(x): 'having the clear block x lying on the table, pick x up by the empty hand, and hold x in the hand',
(iv) `putdown`(x): 'holding block x in the hand, put x down on the table, and leave the hand empty'.

The desired *goal* situation is that a_3 is on the top of a_2.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press



Initial situation Goal situation

Suppose that, *holding block* a_1 *in its hand, the robot intends to put the block* a_1 *on the clear top of block* a_3*, which would leave its hand empty.*

A naive way to *specify* this action $\texttt{stack}(a_1, a_3)$ would be a *Horn axiom* such as:

$$(\texttt{HOLDS}(a_1) \wedge \texttt{CLEAR}(a_3)) \vdash (\texttt{HANDEMPTY} \wedge \texttt{CLEAR}(a_1) \wedge \texttt{ON}(a_1, a_3)). \tag{1}$$

But (1) yields contradictory consequences, such as:

$$(\texttt{HOLDS}(a_1) \wedge \texttt{CLEAR}(a_3)) \vdash ((\texttt{HANDEMPTY} \wedge \texttt{HOLDS}(a_1)) \wedge \texttt{CLEAR}(a_1) \wedge (\texttt{ON}(a_1, a_3) \wedge \texttt{CLEAR}(a_3))). \tag{2}$$

This paradox stems primarily from the fact that our $\wedge$ in question is intended to represent only the *'concurrent co-existence'*, while the traditional semantics of $\wedge$ is too vague so conjunction $\wedge$ can behave differently.

One way out has been proposed by Nilsson's STRIPS (Nilsson 1980; McDermott and Hendler 1995), with changes to both *specification language* and *inference rules*:

(1) Each action α is specified in terms of its *precondition* $Pre(\alpha)$, which consists of predicate-calculus atomic formulas and/or their negations, and two lists of atomic predicate formulas: *add-list* $Add(\alpha)$ and *delete-list* $Del(\alpha)$.
(2) The action specification is applied to *edit* descriptions of situations instead of being used as an *axiom* in *deducing properties* of situations.
Namely, generating a new situation description from an old one is a matter of deleting all the atomic formulas taken from $Del(\alpha)$ and adding all the ones taken from $Add(\alpha)$.

For example, $\alpha := \texttt{stack}(a_1, a_3)$ is specified in STRIPS as:

(a) $Pre(\alpha) := \{\texttt{HOLDS}(a_1), \texttt{CLEAR}(a_3)\}$,
(b) $Del(\alpha) := \{\texttt{HOLDS}(a_1), \texttt{CLEAR}(a_3)\}$,
(c) $Add(\alpha) := \{\texttt{HANDEMPTY}, \texttt{CLEAR}(a_1), \texttt{ON}(a_1, a_3)\}$.

Accordingly, a situation of the form

$$\{\texttt{HOLDS}(a_1), \texttt{CLEAR}(a_3), \texttt{CLEAR}(a_2)\}$$

can be transformed into

$$\{\texttt{HANDEMPTY}, \texttt{CLEAR}(a_1), \texttt{ON}(a_1, a_3), \texttt{CLEAR}(a_2)\}.$$

Another approach, the logical one, is developed in Masseron *et al.* (1993), where the main idea is as follows:

(a) *do not* change the 'naive' Horn-like specification language,
(b) but *refine* the system of logical inference rules so that the AI systems can be properly handled in purely logical terms.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

As a logical setting, Masseron *et al.* (1993) invokes *linear logic* (Girard 1987), a resource-sensitive *refinement* of the traditional logic. One reason is that the traditional $\wedge$ is split within linear logic into two connectives: $\otimes$ (*tensor*) and & (*with*), and, dually, the traditional $\vee$ is split into two connectives: $\parr$ (*par*) and $\oplus$ (*plus*).

(i) The intended meaning of $A \otimes B$ is *'A and B co-exist'*,
(ii) whereas $A \& B$ is intended to represent *'any of A and B'*.

In particular, $A \otimes A$ specifies the situation when the *two copies* of A co-exist, and generally $A \otimes A \neq A$.

Dually to &, $A \oplus B$ is intended to represent *'either A or B'*, which will help us to describe actions with *non-deterministic* effects (see Example 2.2).

Based on the *'co-existence'* essence of $\otimes$, the action $\mathtt{stack}(a_1, a_3)$ in question is specified in the same *naive* way as (1):

$$(\mathtt{HOLDS}(a_1) \otimes \mathtt{CLEAR}(a_3)) \vdash (\mathtt{HANDEMPTY} \otimes \mathtt{CLEAR}(a_1) \otimes \mathtt{ON}(a_1, a_3)). \tag{3}$$

But now in linear logic the corresponding $\otimes$-version of the undesired (2) is not derivable from (3).

Although the approach of Masseron *et al.* (1993) is logically justified, we run into difficulties with the complexity bounds of the planning problems related to the reachability problem for Petri nets (Gehlot and Gunter 1990) and the undecidability of Horn fragments of linear logic (Kanovich 1995).

Now we will introduce simple syntactical constraints that allow us to obviate the problem of the overestimated complexity bounds, with the advantage that the well-known AI planning problems can be reformulated under these syntactical requirements in a natural way. Furthermore, our syntactical constraints are automatically provided by the proper choice of the most *primitive* basic predicates. For instance, in Example 1.1, the original $\mathtt{HOLDS}(a_1)$ incorporates a number of more *primitive* correlations: when the robot *picks up* block a_1 whose top is *clear*, among other things it *changes* the status of the top of a_1; since there is no information about the design peculiarities of the hand, the only thing we can say is that the status of the top of a_1 becomes *uncertain*. Action $\mathtt{stack}(a_1, a_3)$ *changes*, in particular, the status of the top of a_1 from *uncertain* to *clear*. As a result, we get the following *well-balanced* reformulation of Axiom (3)

$$(\mathtt{holds}(a_1) \otimes \mathtt{uncertain}(a_1) \otimes \mathtt{clear}(a_3)) \vdash (\mathtt{handempty} \otimes \mathtt{clear}(a_1) \otimes \mathtt{on}(a_1, a_3)). \tag{4}$$

In addition, $\mathtt{HOLDS}(a_1)$ includes the fact that a_1 is *not* on the table.
The above argument explains our choice of the following *primitive* predicates:

(a) $\mathtt{on}(x, y)$ means *'block x is on the top of block y'*,
(b) $\mathtt{clear}(x)$ stands for *'no block is on the top of block x'*,
(c) $\mathtt{uncertain}(x)$ stands for *'the status of the top of block x is uncertain'*,
(d) $\mathtt{handempty}$ means *'the robot's hand is empty'*,
(e) $\mathtt{holds}(x)$ stands for *'the hand holds block x'*,
(f) $\mathtt{ontable}(x)$ means *'the bottom of block x is on the table'*, and, finally,
(g) $\mathtt{notontable}(x)$ stands for *'block x is not on the table'*.

These provide a linear logic specification of the system in terms of *well-balanced* Horn axioms like (4).

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

Our approach allows us to handle the planning aspects of AI systems in purely logical terms, where the difficulties of being over-complicated are obviated within the framework of a *Horn fragment* of linear logic in the sense of Kanovich (1992).

The Horn fragment used here operates with *pure Horn clauses* of the form $X \vdash Y$ and *branching Horn clauses* of the form $X \vdash (Y_1 \oplus \cdots \oplus Y_\ell)$, where the X and Y's are built up from atomic predicate expressions $Q(t_1, .., t_k)$ only by $\otimes$.

At the very beginning, we use the Horn linear logic as a *specification language* for describing the robot systems in question. The actions performed by a robot are specified in terms of Horn axioms, *branching* Horn axioms are used for specifying the actions with *non-deterministic effects*.

After that, the *planning* problem is represented as a sequent of the form $W \vdash \widetilde{Z}$, where W specifies a given initial situation, and $\widetilde{Z}$ describes the desired set of final partial situations (Masseron *et al.* 1993).

Our approach to *making plans* includes two steps:

(a) *First,* we attempt to *derive* $W \vdash \widetilde{Z}$ within linear logic theory T that consists of those axioms specifying the possible actions of the robot.

(b) *Second,* if we have succeeded, we *extract* a plan of the actions from the derivation found.

To justify our approach:

(a) *First,* the existence of a plan is proved to be equivalent to derivability of the corresponding $W \vdash \widetilde{Z}$.

(b) *Second,* the *exact* correspondence is established between arbitrary solutions to the planning problem and proofs developed within T for $W \vdash \widetilde{Z}$.

As for the complexity of the planning problem, the situation seems to be discouraging. This is because Kanovich (1995) provides us with an LL theory T_0 over a *fixed* number of atomic propositions, which consists of a *fixed* number of axioms of the form $p \otimes q \vdash r$ or $p \vdash q \otimes r$ or $p \vdash q \oplus r$ (where p, q, r are atomic propositions), and the problem of determining for any sequent of the form $p \otimes p \otimes \cdots \otimes p \vdash q$ whether this sequent is derivable within T_0 is *undecidable*. According to Gehlot and Gunter (1990) we meet with essential difficulties of the reachability problem for general Petri nets, even if we confine ourselves to pure Horn axioms.

Nevertheless, we find a reasonably wide class of the *natural* robot systems, which can be axiomatized in such a way that their planning problems can be efficiently resolved.

The *main assumption* underlying the specification of our class of robot systems is that each of the actions obeys a certain *law of conservation* such as *'the number of the primitive correlations newly established is equal to the number of the primitive correlations destroyed in the process of the action'*.

According to what have been seen in Example 1.1, the most primitive ('indivisible') relations are a good choice. As a rule, the primitive correlations ('indivisible' *facts*) form certain *modules*, or *units* (*cf.* Nilsson (1980)). Here we use *unit* to mean a set of 'disjoint' closed primitive correlations such that at any moment *exactly one* of these correlations holds.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

(a) Thus, the set:

$$\{\texttt{handempty},\ \texttt{holds}(a_1),\ \texttt{holds}(a_2),\ \ldots,\ \texttt{holds}(a_N)\},$$

forms a unit related to the robot's hand.

(b) Each of the following units is related to the status of the top of the corresponding a_j $(j = 1, 2, ..N)$:

$$\{\texttt{uncertain}(a_j),\ \texttt{clear}(a_j),\ \texttt{on}(a_1, a_j),\ \texttt{on}(a_2, a_j),\ \ldots,\ \texttt{on}(a_N, a_j)\}.$$

(c) Trivial units are related to the table $(j = 1, 2, ..N)$:

$$\{\texttt{ontable}(a_j),\ \texttt{notontable}(a_j)\}.$$

The latter show how to circumvent the problems of *negation.*

Any *consistent* action must *respect* such a unit. For instance, because of *consistency*, no two distinct facts from one and the same unit can be included in either the *precondition* or *postcondition* of a consistent action. This observation makes the main assumption trivial for any robot system with consistent actions whose system of closed atomic correlations can be partitioned into *units.*

The advantage of the linear logic approach is that in the case of the consistent actions, whatever plan extracted from linear logic proofs we take, each of its intermediate steps is *realizable*, and each of the intermediate situations is *consistent*, whenever an initial situation is consistent.

We prove EXPTIME-completeness for deciding whether there is a solution to a planning problem in a given robot system that assumes actions with *non-deterministic effects.*

As for making plans, we meet with the problem of hyper-exponential tree-like plans. However, we design an exptime algorithm to make plans but in a folded form.
As numerical parameters in our exptime algorithm, we invoke two kinds of parameters:

(1) the *length of description* of a planning task 'from W to $\widetilde{Z}$', and

(2) the *length of description* of $\texttt{Ax}_T$, a set of possible actions.

Let us emphasize that the latter parameter, the *length of description* of the theory T of the robot system itself, is accounted in an *honest* way. See details in Section 5.1, where the formal concept of the *honest* descriptive complexity of theories is introduced as a version of Kolmogorov's concept of the algorithmic amount of information (Kolmogorov 1965).

Fixing a finite signature Σ, that is, a finite set of predicates and their finite domains, we observe the phenomenon that, for any robot system over signature Σ whose set of actions is described in polytime, the problem of making plans can be resolved in *polynomial time* with respect to the length of description of a planning task.

The decision complexity is reduced to PSPACE-completeness for the robot systems with only pure *deterministic* actions. The troubles encountered when making plans that are necessarily of exponential length are circumvented by designing a procedure to generate such long plans step-by-step in polynomial space.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

2. Basic definitions

Definition 2.1. A *robot system*, or an *action system*, is defined as in Masseron *et al.* (1993).

The basic concepts of our approach are illustrated using the following well-known examples.

Example 2.1. (**Towers of Hanoi**) A one-handed robot deals with N plates a_1, .., a_n, a_{n+1}, .., a_N of increasing size, and three boxes B_1, B_2, B_3.

Initially, the plates are stacked inside B_1 in the natural order, the largest plate being the bottom one.

The robot is to move the first n plates from B_1 to B_2, one plate at a time, in such a way that no plate is ever placed on a smaller one. Box B_3 may be used for temporary storage of plates.

We formalize the problem with the help of the following predicates:

(a) $\texttt{handempty}$ means *'the robot's hand is empty'*,
(b) $\texttt{holds}(x)$ stands for *'the hand holds plate x'*,
(c) $\texttt{on}(x, y)$ means *'plate x is on plate y'*,
(d) $\texttt{clear}(x)$ stands for *'no plate is on plate x'*,
(e) $\texttt{in}(x, z)$ means *'plate x is inside box z (on its bottom)'*, and
(f) $\texttt{free}(z)$ stands for *'no plates are inside box z'*.

To design a 'tower-moving' theory $\texttt{Hanoi}_N$ along the lines of the above *naive* technique, we specify the actions that the robot makes to move plates with its hand, that is the action of either taking plate a_i, or putting plate a_i somewhere, by the following Horn axioms $(1 \leqslant i < j \leqslant N,\ k = 1, 2, 3)$:

$\texttt{take}(a_i, a_j) :=$

$$(\texttt{handempty} \otimes \texttt{clear}(a_i) \otimes \texttt{on}(a_i, a_j)) \vdash (\texttt{holds}(a_i) \otimes \texttt{clear}(a_i) \otimes \texttt{clear}(a_j)), \qquad (5)$$

$\texttt{Take}(a_i, B_k) :=$

$$(\texttt{handempty} \otimes \texttt{clear}(a_i) \otimes \texttt{in}(a_i, B_k)) \vdash (\texttt{holds}(a_i) \otimes \texttt{clear}(a_i) \otimes \texttt{free}(B_k)), \qquad (6)$$

$\texttt{put}(a_i, a_j) :=$

$$(\texttt{holds}(a_i) \otimes \texttt{clear}(a_j)) \vdash (\texttt{handempty} \otimes \texttt{on}(a_i, a_j)) \quad \underline{(\text{here } i < j)}, \qquad (7)$$

$\texttt{Put}(a_i, B_k) :=$

$$(\texttt{holds}(a_i) \otimes \texttt{free}(B_k)) \vdash (\texttt{handempty} \otimes \texttt{in}(a_i, B_k)). \qquad (8)$$

We have to *make a plan* of the actions to be performed by the robot so that the robot

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

can get to the final situation $\mathtt{Goal}_n$:

$$\mathtt{Goal}_n := (\mathtt{handempty} \otimes \mathtt{clear}(a_1) \otimes \bigotimes_{i=1}^{n-1} \mathtt{on}(a_i, a_{i+1}) \otimes \mathtt{clear}(a_{n+1}) \otimes \mathtt{in}(a_n, B_2) \otimes \mathtt{free}(B_3)),$$

if it started with the initial situation $\mathtt{Init}_n$:

$$\mathtt{Init}_n := (\mathtt{handempty} \otimes \mathtt{clear}(a_1) \otimes \bigotimes_{i=1}^{n-1} \mathtt{on}(a_i, a_{i+1}) \otimes \mathtt{on}(a_n, a_{n+1}) \otimes \mathtt{free}(B_2) \otimes \mathtt{free}(B_3)).$$

Following our approach to such a task,

— *first,* we are looking for a derivation within $\mathtt{Hanoi}_N$ for a sequent of the form

$$\mathtt{Init}_n \vdash \mathtt{Goal}_n$$

that represents the problem, and,

— *second,* we transform the derivation found into a plan of actions.

For $n=2$, the result is shown in Figure 1, where the following C_1, .., C_5 represent 'intermediate situations':

$C_1 := (\mathtt{holds}(a_1) \otimes \mathtt{clear}(a_1) \otimes \mathtt{clear}(a_2) \otimes \mathtt{clear}(a_3) \otimes \mathtt{in}(a_2, B_2) \otimes \mathtt{free}(B_3))$,
$C_2 := (\mathtt{handempty} \otimes \mathtt{clear}(a_1) \otimes \mathtt{clear}(a_2) \otimes \mathtt{clear}(a_3) \otimes \mathtt{in}(a_2, B_2) \otimes \mathtt{in}(a_1, B_3))$,
$C_3 := (\mathtt{holds}(a_2) \otimes \mathtt{clear}(a_1) \otimes \mathtt{clear}(a_2) \otimes \mathtt{clear}(a_3) \otimes \mathtt{free}(B_2) \otimes \mathtt{in}(a_1, B_3))$,
$C_4 := (\mathtt{handempty} \otimes \mathtt{clear}(a_1) \otimes \mathtt{clear}(a_2) \otimes \mathtt{on}(a_2, a_3) \otimes \mathtt{free}(B_2) \otimes \mathtt{in}(a_1, B_3))$,
$C_5 := (\mathtt{holds}(a_1) \otimes \mathtt{clear}(a_1) \otimes \mathtt{clear}(a_2) \otimes \mathtt{on}(a_2, a_3) \otimes \mathtt{free}(B_2) \otimes \mathtt{free}(B_3))$.

We can see a nice correspondence between the 'Horn-like' derivation and the plan shown there.

Example 2.2. (**Socks**) A robot deals with a heap of N socks, either black or white, hidden in a dark room.

The challenge is to pick out enough socks to be sure to obtain n one-coloured socks.
We formalize the sock problem by means of the following propositions:

(a) H stands for *'a sock is hidden'*,
(b) W means *'a sock turned out to be white'*, and
(c) B means *'a sock turned out to be black'*.

Our theory $\mathtt{Sock}_N$ includes only one proper axiom.
The action of *picking out a sock* is specified by the axiom:

$$H \vdash (W \oplus B), \tag{9}$$

the *branching* $\oplus$-form of which puts great emphasis on the fact that the effect of this action is *non-deterministic*: the robot does not know, *in advance*, what the colour of the sock being picked out will be.

The *initial situation* will be represented as H^N.
Here, and henceforth, we set

$$H^N := \underbrace{H \otimes H \otimes \cdots \otimes H}_{N \text{ times}}.$$

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

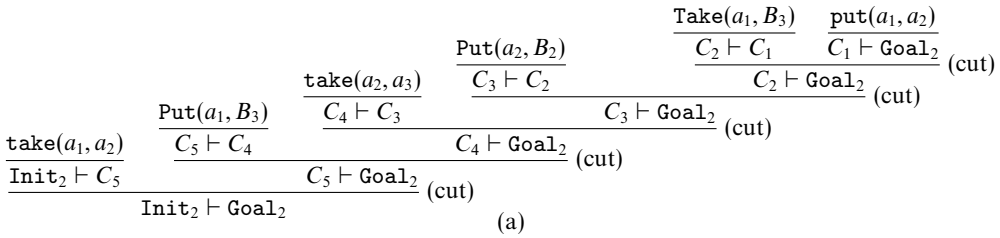


Fig. 1. The derivation (a) and plan (b) for moving 2 plates by a one-handed robot.

The advantage of linear logic is that such an H^N describes the situation when *exactly N copies* of a sock are hidden in the room, whereas the traditional

$$H \wedge H \wedge \cdots \wedge H$$

coincides with H.

The *planning* problem is that the robot, starting with H^N, is to get to the situations that 'contain' either W^n or B^n as a *partial goal.*
We will specify this task by a sequent of the form

$$H^N \vdash ((\bigoplus_j (W^n \otimes V_{1,j})) \oplus (\bigoplus_k (B^n \otimes V_{2,k})))$$

where by means of elementary products $V_{1,1},..., V_{1,t_1}$, $V_{2,1},..., V_{2,t_2}$, we describe possible *environments* of our partial goals, respectively.

According to our approach, *first,* we attempt to prove such a sequent for some $V_{i,j}$ and, *second,* we extract a plan from the proof we have found.

For $N=3$, $n=2$, the result is shown in Figure 2. The plan there represents a *winning strategy* that envisages *all* possible reactions of Nature on the road from the initial situation to either final one. Each vertex v prescribes that the robot perform a certain action: its outgoing edges show all its effects. At the root the robot starts by picking out a sock. The *non-deterministic* effect of this action is represented with the help of two

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

outgoing edges labelled by $H \vdash W$ or $H \vdash B$ in accordance with the colour of the sock picked out, etc.

Following the recommendations of the plan, the robot can never ever be punished by its choice of the colour of the hidden socks.

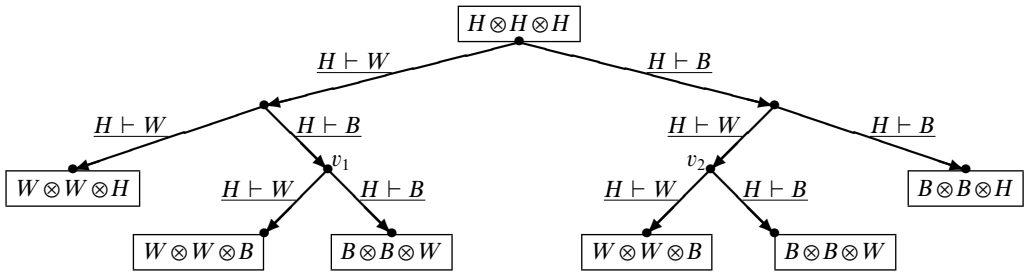
In Figure 2 we also show a *folded* version of the plan obtained by 'gluing' vertices v_1 and v_2, for which

$$\mathtt{value}_{v_1}(H^3) \simeq \mathtt{value}_{v_2}(H^3) \simeq W \otimes B \otimes H,$$

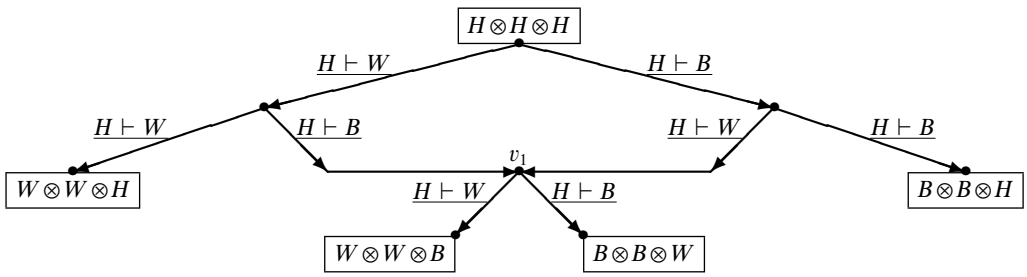
with one of the corresponding subtrees being thrown away.

The sock planning problem turns out to be solvable iff $n \leqslant \frac{N+1}{2}$.

For instance, in the case where $2n-2=N$, the robot would have failed if it had put in the dark room a heap of $n-1$ white socks and $n-1$ black socks.



(a) - in a tree-like form



(b) - in a folded form

Fig. 2. The plan as to pick out a pair of one-coloured socks from 3 socks.

3. Horn-like theories

Definition 3.1. An *LL theory* T is specified by means of a recursive set of its 'proper axioms' (denoted by $\mathtt{Ax}_T$).

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

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

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

A *cut-free proof in a theory* T is a proof within T in which each of the cut-formulas occurs in some proper axiom of T.

We will use the following syntactical conventions.

Definition 3.2. A *signature* Σ consists of predicate symbols $Q_1, Q_2, .., Q_j, \ldots$ with their arity and (many-sorted) constants $d_1, d_2, .., d_i, \ldots$.

An *atomic formula* is of the form $Q_j(t_1, t_2, .., t_k)$, where each of these *terms* t_i is either a variable or a constant. (An infinite list of (many-sorted) variables $x_1, x_2, .., x_i, \ldots$ is assumed.) A closed atomic formula is called a *fact*.

An *elementary product* X of $\otimes$-*rank* m is defined as an expression of the form

$$X := (P_1(t_{1,1}, .., t_{1,k_1}) \otimes P_2(t_{2,1}, .., t_{2,k_2}) \otimes \cdots \otimes P_m(t_{m,1}, .., t_{m,k_m})),$$

where P_1,P_2,..,P_m are predicate symbols, and $t_{1,1}$,..,t_{1,k_1},. . . ,$t_{m,1}$,..,t_{m,k_m} are (many-sorted) terms.

It is convenient to allow *trivial* elementary products V of $\otimes$-rank 0, for which we have $X \otimes V = X$.

We say that elementary products X and X' are *equivalent*, and write

$$X \simeq X'$$

if X does not differ from X' modulo commutativity and associativity of $\otimes$.

A *pure Horn clause* is defined as a sequent of the form

$$X \vdash Y, \tag{10}$$

where X and Y are elementary products.

Generally, a *branching Horn clause* is a sequent of the form

$$X \vdash (Y_1 \oplus Y_2 \oplus \cdots \oplus Y_\ell), \tag{11}$$

where X, Y_1, Y_2,. . . ,Y_ℓ are elementary products ($\ell \geqslant 1$).

If the $\otimes$-rank of each of these $Y_1, Y_2, .., Y_\ell$ is bounded by $\otimes$-rank of the left-hand X, then the Horn clause is called *non-lengthening*. Similarly, we define *non-shortening* Horn clauses.

A Horn clause of the form (11) is said to be *well-balanced* iff $\otimes$-rank of each of these $Y_1, Y_2, .., Y_\ell$ is equal to $\otimes$-rank of X.

A *pure Horn LL theory* is an LL theory in which each of its proper axioms is a pure Horn clause.

An LL theory T such that each of its proper axioms is a branching Horn clause is called a *branching Horn LL theory*.

For AI systems with pure deterministic actions, a plan $\mathscr{P}$ is defined as a chain of the actions (Nilsson 1980; Bibel 1986; Chapman 1987; Masseron *et al.* 1993). In order to cover the general case in which the actions with *non-deterministic effects* are allowed, we extend this definition to tree-like plans $\mathscr{P}$, in which each vertex v prescribes the performance of a certain action, with its outgoing edges showing all possible reactions of Nature.

Definition 3.3. Let $\mathtt{Ax}_T$ be a set of Horn clauses specifying actions in a given robot system.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

A labelled directed tree $\mathscr{P}$, in which each edge is labelled by a pure Horn clause, is said to be a *tree-like plan based on* $\mathtt{Ax}_T$ if for each vertex v with exactly $\ell \geqslant 1$ sons w_1, w_2, .., w_ℓ, the outgoing edges (v, w_1), (v, w_2), .., (v, w_ℓ) are labelled by pure Horn clauses with one and the same antecedent: $X \vdash Y_1$, $X \vdash Y_2$, .., $X \vdash Y_\ell$, respectively – these clauses being such that there exists a Horn clause of the form $X \vdash (Y_1 \oplus Y_2 \oplus \cdots \oplus Y_\ell)$ that belongs to $\mathtt{Ax}_T$. (Thus we represent all possible effects of the action).

The root of $\mathscr{P}$ is conceived of as the *input* vertex. A leaf in $\mathscr{P}$ is specified as an *output* vertex.

Let an elementary product W be assigned to the root v_0: by definition,

$$\mathtt{value}_{v_0}(W) := W.$$

By induction, we assign $\mathtt{value}_v(W)$ to each vertex v as follows.

(a) For any edge (v, w) labelled by $X \vdash Y$, if $\mathtt{value}_v(W)$ is defined and $\mathtt{value}_v(W) \simeq (X \otimes V)$ for some V, then

$$\mathtt{value}_w(W) := (Y \otimes V).$$

(b) Otherwise, $\mathtt{value}_w(W)$ is declared to be undefined.

Given an *initial* W and a set of *exact goals* $\{Z_1, .., Z_k\}$, a plan $\mathscr{P}$ based on $\mathtt{Ax}_T$ is said to be *exactly leading* from W to $\{Z_1, .., Z_k\}$ if for each of the leaves v of $\mathscr{P}$, $\mathtt{value}_v(W)$ is defined and for some j such that $1 \leqslant j \leqslant k$,

$$\mathtt{value}_v(W) \simeq Z_j.$$

Given an *initial* W and a set of *partial goals* $\{Z_1, .., Z_k\}$, we will say that a plan $\mathscr{P}$ based on $\mathtt{Ax}_T$ *leads* from W to $\{Z_1, .., Z_k\}$ if for each of the leaves v of $\mathscr{P}$, $\mathtt{value}_v(W)$ is defined and there is a j such that $1 \leqslant j \leqslant k$ and for some elementary product V,

$$\mathtt{value}_v(W) \simeq (Z_j \otimes V).$$

Comment 3.1. There is *no* correlation between the number of exact (partial) goals k and the number of leaves in plans $\mathscr{P}$. For instance, the set of goals $\{Z_1, .., Z_k\}$ might have been expanded to any set of goals $\{Z_1, .., Z_k, Z\}$.

A plan $\mathscr{P}$ *leads* from W to a set of *partial goals* $\{Z_1, .., Z_k\}$ if and only if for some elementary products $V_{1,1}, \ldots, V_{1,t_1}, \ldots, V_{k,1}, \ldots, V_{k,t_k}$, plan $\mathscr{P}$ *exactly leads* from W to the set of goals $\{(Z_1 \otimes V_{1,1}), .., (Z_1 \otimes V_{1,t_1}), \ldots, (Z_k \otimes V_{k,1}), .., (Z_k \otimes V_{k,t_k}\}$.

We can express the *frame problem* from Nilsson (1980) as follows. When a robot processes a plan in the case of *exact* goals, any *environment* V remains *unchangeable*: a plan $\mathscr{P}$ *exactly leads* from W to $\{Z_1, .., Z_k\}$ iff for any V, plan $\mathscr{P}$ *exactly leads* from $(W \otimes V)$ to $\{(Z_1 \otimes V), .., (Z_k \otimes V)\}$. A similar phenomenon of *unchangeable* environments occurs in the case of *partial* goals.

3.1. *Exact goals*

Theorem 3.1. (Completeness) (*cf.* Kanovich (1992)) For a given branching Horn LL theory T, let $\mathscr{P}$ be a finite tree-like plan based on $\mathtt{Ax}_T$ that exactly leads from W to $\{Z_1, .., Z_k\}$.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

Then a sequent of the form

$$W \vdash (Z_1 \oplus \cdots \oplus Z_k)$$

is LL-provable within T.

Proof. Passing through the tree $\mathscr{P}$ from its leaves to its root, we assemble the corresponding LL-proof within T.

Since $Z \oplus Z = Z$, we contract the Z_i repeated too many times (*cf.* Comment 3.1). □

Theorem 3.2. (Soundness) Let W and $Z_1,..,Z_k$ be elementary products such that

$$W \vdash (Z_1 \oplus \cdots \oplus Z_k)$$

is LL-provable within a branching Horn LL theory T.

Then there exists a finite tree-like plan $\mathscr{P}$ based on $\mathtt{Ax}_T$ that exactly leads from W to $\{Z_1, .., Z_k\}$, and, in addition to that, there are no distinct vertices v_1 and v_2 on a path in $\mathscr{P}$ such that $\mathtt{value}_{v_1}(W) \simeq \mathtt{value}_{v_2}(W)$.

Proof. The main idea is as follows (*cf.* Kanovich (1992)).

Given an LL-proof within T for a sequent of the form $W \vdash (Z_1 \oplus \cdots \oplus Z_k)$, we transform it into a cut-free LL-proof within T.

Then, by induction on this cut-free derivation, starting with its root, we assemble a finite tree-like plan $\mathscr{P}_0$ based on $\mathtt{Ax}_T$ so that $\mathscr{P}_0$ exactly leads from W to $\{Z_1, .., Z_k\}$. (See, for instance, Figure 1)

It remains to remove repetitions. Suppose for some vertex v_1 and its proper *descendant* v_2 that

$$\mathtt{value}_{v_1}(W) \simeq \mathtt{value}_{v_2}(W).$$

Then we contract $\mathscr{P}_0$ to some $\mathscr{P}_1$ as follows. We take the subtree whose root is v_2, and substitute it for the subtree whose root is v_1. The contracted tree $\mathscr{P}_1$ is shown to be also a plan based on $\mathtt{Ax}_T$, exactly leading from W to $\{Z_1, .., Z_k\}$.

By repeatedly applying this process, we eventually obtain the desired plan $\mathscr{P}$ that contains no redundant moves. □

For LL theories T with non-lengthening axioms, we can contract the *search space* for plans as follows.

Theorem 3.3. Let T be a branching Horn LL theory such that all its proper axioms are non-lengthening, and let W be a closed elementary product of $\otimes$-rank m.

Any finite plan based on $\mathtt{Ax}_T$ that exactly leads from W to $\{Z_1, .., Z_k\}$, where $Z_1,..,Z_k$ are closed elementary products, can be transformed into a finite tree-like plan $\mathscr{P}$ based on $\mathtt{Ax}_T$ such that $\mathscr{P}$ exactly leads from W to $\{Z_1, .., Z_k\}$, and, in addition:

(a) in each of the axioms involved inside $\mathscr{P}$ the $\otimes$-rank of each of its $\otimes$-terms is also bounded by m, and,
(b) the height of $\mathscr{P}$ is bounded by $L_T(m)$, the total number of all non-equivalent closed elementary products whose $\otimes$-rank does not exceed m.

Proof. Given a plan leading from W to $\{Z_1, .., Z_k\}$, by Theorems 3.2 and 3.1, we

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

transform it first into an LL-proof and then into a tree-like plan $\mathscr{P}$ so that all $\mathtt{value}_v(W)$ are non-equivalent on each of the branches of $\mathscr{P}$.

Since each of the Horn clauses the edges of $\mathscr{P}$ are labelled by is non-lengthening, the $\otimes$-rank of each of these $\mathtt{value}_v(W)$ does not exceed m, either. Therefore, $L_T(m)$ is an upper bound of the *height* of $\mathscr{P}$.

It should be pointed out that $L_T(m)$ becomes *infinite* for the signatures wherein either the number of predicate symbols is infinite, or the number of constants is infinite. □

Corollary 3.1. (Decidability) There is an algorithm applicable to the following pair of inputs:

(i) to any recursive branching Horn LL theory T over a finite signature such that each of the proper axioms of T is non-lengthening, and,
(ii) to any closed elementary products W, Z_1,..,Z_k,

which:

(a) determines whether there is a plan that is based on $\mathtt{Ax}_T$ and is exactly leading from W to $\{Z_1, .., Z_k\}$, and,
(b) if the answer is positive, makes such a plan.

Proof. We will describe the algorithm we need – it investigates only plans that satisfy the constraints from Theorem 3.3.

Input: An algorithm ρ that describes the axioms of T and elementary products W, Z_1,..,Z_k.

Method: Let m be the least upper bound of $\otimes$-ranks of W, Z_1,..,Z_k.

Since the signature of T consists of a finite number of predicate symbols and of a finite number of constants, the total number $L_T(m)$ of all non-equivalent closed elementary products whose $\otimes$-rank does not exceed m is also finite.

Let us list the set $\widehat{\mathscr{P}}$ of all tree-like plans $\mathscr{P}$ based on $\mathtt{Ax}_T$ such that:

(a) in each of the axioms involved inside $\mathscr{P}$ the $\otimes$-rank of each of its $\otimes$-terms is bounded by m, and,
(b) height of $\mathscr{P}$ is bounded by $L_T(m)$.

It is readily seen that $\widehat{\mathscr{P}}$ is finite.

Output: Check each of the plans $\mathscr{P}$ from $\widehat{\mathscr{P}}$, whether $\mathscr{P}$ exactly leads from W to $\{Z_1, .., Z_k\}$ or not.

If the result of the checking procedure is positive for some $\mathscr{P}$ from $\widehat{\mathscr{P}}$, return with this plan $\mathscr{P}$ as a solution to the problem.

Otherwise, return with the negative 'No plans'.

Theorem 3.3 guarantees that the above procedure for making plans is correct. □

Theorem 3.4. The finiteness of the number of constants is a crucial point in the decidability result.

Proof. Assume that we have an infinite number of constants c_1,.., c_n,. . . .

Given a Turing machine M, we will design a polytime-decidable pure Horn LL theory T_0 over the signature that consists of one unary predicate symbol P_1 and the constants c_1,.., c_n,. . . .

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

Suppose an enumeration of all possible configurations of M:

$$K_1, K_2, \ldots, K_n, \ldots.$$

The set $\mathtt{Ax}_{T_0}$ is defined as the set of all

$$P_1(c_i) \vdash P_1(c_j)$$

such that M can go from K_i to K_j in one move.

Theorems 3.1 and 3.2 provide that there is a plan based on $\mathtt{Ax}_{T_0}$ that (exactly) leads from $P_1(c_1)$ to $\{P_1(c_n)\}$ if and only if M can go from K_1 to K_n.

Taking a Turing machine M, with its halting problem being undecidable, we get undecidability for the planning problem over the above infinite domain. □

3.2. *Partial goals*

Lemma 3.1. Let T be a branching LL theory and W be an elementary product.
A sequent of the form

$$W \vdash (Z_1 \oplus \cdots \oplus Z_k)$$

is AL-provable within T if and only if there exist $U_1, \ldots, U_k$ such that

$$W \vdash ((Z_1 \otimes U_1) \oplus \cdots \oplus (Z_k \otimes U_k))$$

is LL-provable within T, where $U_1, \ldots, U_k$ are sums of elementary products: each of these U_i is of the form

$$U_i = V_{i,1} \oplus \cdots \oplus V_{i,t_i}.$$

Proof. The direction from LL-proofs to AL-proofs is immediate as all $(Z_i \otimes U_i) \vdash Z_i$ are provable in *affine logic*.
For the other direction, we prove by an induction on a cut-free AL-proof within T for the sequent

$$\Pi, \Delta \vdash (Z_1 \oplus \cdots \oplus Z_k)$$

that there exist $U_1, \ldots, U_k$ such that

$$\Pi, \Delta \vdash ((Z_1 \otimes U_1) \oplus \cdots \oplus (Z_k \otimes U_k))$$

is LL-provable within T, where Π is a multiset of elementary products and Δ is a multiset of sums of elementary products. □

Corollary 3.2. (Soundness and Completeness) Let T be a branching Horn LL theory, and W, $Z_1,..,Z_k$ be elementary products.
The following three propositions are pairwise equivalent:

(a) A sequent of the form $W \vdash (Z_1 \oplus \cdots \oplus Z_k)$ is AL-provable within T.
(b) There exists a finite plan based on $\mathtt{Ax}_T$ that leads from W to the set of partial goals $\{Z_1, .., Z_k\}$.
(c) There exists a finite tree-like plan $\mathscr{P}$ based on $\mathtt{Ax}_T$ that leads from W to the set of partial goals $\{Z_1, .., Z_k\}$, and, in addition, all $\mathtt{value}_v(W)$ are non-equivalent on each of the branches of $\mathscr{P}$.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

Proof. According to Lemma 3.1, the first item is equivalent to:

— 'There exist $U_1, \ldots, U_k$, with each U_i being of the form $V_{i,1} \oplus \cdots \oplus V_{i,t_i}$, such that the sequent

$$W \vdash ((Z_1 \otimes U_1) \oplus \cdots \oplus (Z_k \otimes U_k))$$

is LL-provable within T.'

Since $\otimes$ is distributive with respect to $\oplus$, by Theorems 3.2 and 3.1, the above is equivalent to:

— 'For some elementary products $V_{1,1},\ldots,V_{1,t_1}, \ldots,V_{k,1},\ldots,V_{k,t_k}$, there exists a finite tree-like plan $\mathscr{P}$ based on $\mathtt{Ax}_T$, which *exactly* leads from W to $\{(Z_1 \otimes V_{1,1}),..,(Z_1 \otimes V_{1,t_1}),\ldots,(Z_k \otimes V_{k,1}),..,(Z_k \otimes V_{k,t_k}\}$'.

Taking into account Theorem 3.3 and Comment 3.1, the latter is equivalent to either of the last items. □

4. Lower bounds

We start with the simulation of alternating Turing machines (ATM) introduced in Chandra *et al.* (1981). An alternating machine M is like a Turing machine with the addition of a function associating one of the labels $\wedge$ or $\vee$ with each non-final state. According to Chandra *et al.* (1981):

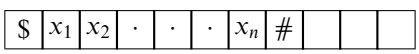
> When in a configuration α with several successors $\beta_1,..,\beta_\ell$, the machine spawns ℓ independent offsprings which run to completion, report acceptance or rejection back to their parent α, and then die. The parent α combines the answers in a simple way (by ANDing the answers if the state q within α is labelled by $\wedge$, or by ORing the answers if this state q is labelled by $\vee$), the resulting answer is passed up to α's parent, and so on.

Theorem 4.1. Any alternating Turing machine M that accepts in space n can be faithfully axiomatized within propositional well-balanced branching Horn LL theories.

Proof. Let M be an alternating machine that accepts in space n.
Without loss of generality, we assume the following:

(a) ATM M has only one tape, which is one-way infinite to the right. The leftmost cell (numbered by 0) contains the marker \$ unerased.

(b) Initially, an *input* string, say $x_1x_2..x_n$, is written in cells 1, 2,..,n on the tape. In addition, a special marker # is written in the $(n+1)$-th cell.



(c) The program of M contains no instruction that could erase either \$ or #. There is no instruction that could move the head of M either to the right when M scans symbol #, or to the left when M scans symbol \$. As a result, M acts in the space between the two unerased markers.

(d) Finally, M has only one *accepting* state, and, moreover, all *accepting* configurations in space n are of one and the same form.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

For each n, we design a theory T_n as follows.
We introduce the following 0-ary predicates (propositions):

(a) $R_{i,\xi}$ stands for *'the i-th cell contains symbol ξ'*,
here $i=0,1,..,n+1$, ξ is a symbol of the tape alphabet of M,
(b) $S_{j,q}$ means *'the j-th cell is scanned by M in state q'*,
here $j=0,1,..,n+1$, q is a state of M.

Given an *instantaneous description* (*configuration*) of M in space n – that M scans the j-th cell in state q, when a string $\xi_0\xi_1\xi_2..\xi_i..\xi_n\xi_{n+1}$ is written left-justified on the otherwise blank tape – we will represent it by a product of the form

$$S_{j,q} \otimes R_{0,\xi_0} \otimes R_{1,\xi_1} \otimes R_{2,\xi_2} \otimes \cdots \otimes R_{n,\xi_n} \otimes R_{n+1,\xi_{n+1}}. \tag{12}$$

Necessarily, $\xi_0 = \$$, and $\xi_{n+1} = \#$.
For each of the *existential* instructions $q\xi \to q'\eta D$, where q is labelled by $\vee$:

— *'if in state q looking at symbol ξ, replace it by η, move the tape head one cell in direction D along the tape, and go into state q''*,

is specified by the set of $n+2$ pure Horn axioms of the form, for $i=0,1,..,n+1$,

$$(S_{i,q} \otimes R_{i,\xi}) \vdash (S_{i_D,q'} \otimes R_{i,\eta}), \tag{13}$$

where $i_D := i+1$ if D is *right*, and $i_D := i-1$ if D is *left*, and $i_D := i$, otherwise.
(Neither $i_{right} = n+2$, nor $i_{left} = -1$ can occur.)
For any state q' labelled by $\wedge$, and any symbol ξ, we collect all instructions with the same left-hand side $q'\xi$:

$$q'\xi \to q_1\eta_1 D_1, \quad \ldots, \quad q'\xi \to q_\ell\eta_\ell D_\ell.$$

The *universal* move of M made by invoking these ℓ instructions:

— *'if in state q' looking at symbol ξ, spawn ℓ independent moves in parallel, each of them is as follows: replace ξ by the corresponding η_i, move the tape head one cell in direction D_i along the tape, and go into state q_i'*,

is specified by the set of $n+2$ *branching* Horn axioms, $i=0,1,..,n+1$:

$$(S_{i,q'} \otimes R_{i,\xi}) \vdash ((S_{i_{D_1},q_1} \otimes R_{i,\eta_1}) \oplus \cdots \oplus (S_{i_{D_\ell},q_\ell} \otimes R_{i,\eta_\ell})). \tag{14}$$

Lemma 4.1. Given an input string $x_1x_2..x_n$ of length n, let W_n be an elementary product of the form (12), which represents the *initial* configuration with the input string $x_1x_2..x_n$, and let Z_n be an elementary product of the form (12), which represents the unique *accepting* configuration of length n.
The following propositions are pairwise equivalent:

(a) The input string $x_1x_2..x_n$ is accepted by M.
(b) A sequent of the form $W_n \vdash Z_n$ is LL-provable within T_n.
(c) A sequent of the form $W_n \vdash Z_n$ is AL-provable within T_n.
(d) There exists a finite plan based on $\mathtt{Ax}_{T_n}$ that (exactly) leads from W_n to $\{Z_n\}$.

Proof. Our encoding is proceeding in a straight course, so that any successful alternating computation in space n that leads from the initial configuration represented by W_n to

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

the accepting configuration represented by Z_n, is immediately re-written as an LL-proof within T_n for $W_n \vdash Z_n$.

The most complicated direction is that such a straightforward encoding is *faithful*.

Suppose that $W_n \vdash Z_n$ is AL-provable within T_n.

By Lemma 3.1, for some $V_1,..,V_t$, a sequent of the form

$$W \vdash ((Z_n \otimes V_1) \oplus \cdots \oplus (Z_n \otimes V_t))$$

is LL-provable within T_n.

By Theorem 3.2, there is a finite tree-like plan $\mathscr{P}$ based on $\mathtt{Ax}_{T_n}$ that exactly leads from W_n to $\{(Z_n \otimes V_1), \ldots, (Z_n \otimes V_t)\}$.

Since each of the axioms (13) and (14) is *well-balanced*, and $\otimes$-rank of both W_n and Z_n is just the same $n+3$, each of these $V_1,..,V_t$ must be *trivial*, which means that for any leaf w in $\mathscr{P}$, the following holds:

$$\mathtt{value}_w(W_n) = Z_n.$$

On the other hand, because of the specific form of our axioms (13) and (14), any $\mathtt{value}_v(W_n)$ in $\mathscr{P}$ is of the form (12), and, hence, represents a configuration of M in space n.

Passing through the tree $\mathscr{P}$ from its leaves to its root v_0, we prove that whatever vertex v we take, there is a successful alternating computation performed by M leading from the configuration represented by $\mathtt{value}_v(W_n)$ to the *accepting* configuration represented by Z_n.

In particular, since $\mathtt{value}_{v_0}(W_n) = W_n$, we can conclude that the given input string $x_1x_2..x_n$ is accepted by M. □

Comment 4.1. The signature of T_n consists only of $O(n)$ 0-ary predicate symbols – the number of the constants invoked is zero.

As for the descriptive complexity of T_n, the list of all proper axioms of T_n can be easily generated in time and space approximately:

$$(n+2) \times (\text{the number of instructions in the program of } M).$$

Furthermore, the set of all proper axioms of T_n can be recognized in `polytime`, based only on two elements of information:

(a) the program of M (which is constant), and
(b) the binary code of the number n (which is variable).

The effect is that the *honest* descriptive complexity of T_n does not exceed

$$\mathrm{Const}_M + \log_2 n.$$

Corollary 4.1. (EXPTIME-hardness)

(a) There is a `polytime` sequence of propositional LL theories T_n (each $\mathtt{Ax}_{T_n}$ consists of $O(n)$ well-balanced two-branching Horn axioms of $\otimes$-rank 2) such that the problem
   — of deciding, for any n and any sequent of the Horn form $W \vdash Z$ such that both W and Z are of $\otimes$-rank $n+2$, whether this sequent is AL-derivable within T_n
   is `EXPTIME`-hard.
(b) Given a finite domain, there is a `polytime` sequence of action systems $\mathscr{A}_n$ over this

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

fixed domain (each of $\mathscr{A}_n$ consists of $O(n)$ well-balanced actions with at most two non-deterministic effects) such that the planning problem

— of deciding, for any n and for any initial situation W and any partial goal Z, whether there exists a plan $\mathscr{P}$ based on the actions from $\mathscr{A}_n$ that leads from W to $\{Z\}$

is EXPTIME-hard.

Proof. First, we construct an exptime deterministic Turing machine $\mathscr{U}$ so that, for any positive integer c and for any deterministic Turing machine M that accepts in time $2^{c \cdot n}$, there exists a binary string κ_M such that, whatever input string $x = x_1 x_2 .. x_m$ we take, x is accepted by M iff $\mathscr{U}$ accepts a string of the form

$$c * \kappa_M * x * \underbrace{111111\ldots1}_{c|x| \text{ times}}.$$

It is readily seen that the acceptance problem for the 'universal' $\mathscr{U}$ is EXPTIME-complete.

According to Chandra *et al.* (1981), our $\mathscr{U}$ is simulated by an alternating machine that accepts in polynomial space, and, moreover, the branching degree of universal moves performed by the ATM there is at most 4, which can be easily reduced to 2.

In turn, the latter ATM is polytime reduced to an alternating machine $M_{\mathscr{U}}$ that accepts in space n.

Finally, we take the theories T_n from the proof of Theorem 4.1 designed for this fixed ATM $M_{\mathscr{U}}$.

The second item is a reformulation of the first one, based on Corollary 3.2. □

Comment 4.2. Both Theorem 4.1 and Corollary 4.1 deal with the *pure propositional* case. The signature of each of our T_n includes only an *unbounded* number of predicate symbols (of 0-ary predicates), and includes no constants at all.

If we confine ourselves both to a fixed number of predicate symbols and to a fixed number of constants, Theorem 4.1 fails because of the *polytime* upper bound of Corollary 5.1.

Nevertheless, we get EXPTIME-hardness also at the opposite end of the 'scale' *with a fixed number of predicate symbols and an unbounded number of constants.*

Theorem 4.2. Any alternating Turing machine M that accepts in space n can be faithfully axiomatized within well-balanced branching Horn LL theories with a fixed number of unary predicates.

Proof. For each n, we will design a theory T'_n by modifying the construction of Theorem 4.1 as follows.

Assume a list of constants:

$$c_0, c_1, c_2, \ldots, c_n, c_{n+1}.$$

We introduce the following unary predicate symbols:

(a) $\widehat{R}_\xi$, for every ξ from the tape alphabet of M.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

The intended meaning of $\widehat{R}_\xi$ is:

$$\widehat{R}_\xi(c_i) = R_{i,\xi}.$$

(b) $\widehat{S}_q$, for every q from the set of states of M.

The intended meaning of $\widehat{S}_q$ is:

$$\widehat{S}_q(c_j) = S_{j,q}.$$

Given an *instantaneous description* (*configuration*) of M in space n – that M scans j-th cell in state q, when a string $\xi_0\xi_1\xi_2..\xi_i..\xi_n\xi_{n+1}$ is written left-justified on the otherwise blank tape – we will now represent it by (*cf.* (12))

$$\widehat{S}_q(c_j) \otimes \widehat{R}_{\xi_0}(c_0) \otimes \widehat{R}_{\xi_1}(c_1) \otimes \widehat{R}_{\xi_2}(c_2) \otimes \cdots \otimes \widehat{R}_{\xi_n}(c_n) \otimes \widehat{R}_{\xi_{n+1}}(c_{n+1}). \tag{15}$$

Each of the axioms of the form (13) is rewritten as

$$(\widehat{S}_q(c_i) \otimes \widehat{R}_\xi(c_i)) \vdash (\widehat{S}_{q'}(c_{i_D}) \otimes \widehat{R}_\eta(c_i)). \tag{16}$$

Each of the branching axioms of the form (14) is rewritten as

$$(\widehat{S}_{q'}(c_i) \otimes \widehat{R}_\xi(c_i)) \vdash ((\widehat{S}_{q_1}(c_{i_{D_1}}) \otimes \widehat{R}_{\eta_1}(c_i)) \oplus \cdots \oplus (\widehat{S}_{q_\ell}(c_{i_{D_\ell}}) \otimes \widehat{R}_{\eta_\ell}(c_i))). \tag{17}$$

□

The *faithfulness* of the modified encoding is established by the following lemma.

Lemma 4.2. Given an input string $x_1x_2..x_n$ of length n, let W_n be an elementary product of the form (15) that represents the *initial* configuration with the input string $x_1x_2..x_n$, and let Z_n be an elementary product of the form (15) that represents the unique *accepting* configuration of length n.

The following propositions are pairwise equivalent:

(a) The input string $x_1x_2..x_n$ is accepted by M.
(b) A sequent of the form $W_n \vdash Z_n$ is LL-provable within T'_n.
(c) A sequent of the form $W_n \vdash Z_n$ is AL-provable within T'_n.
(d) There exists a finite plan based on $\mathtt{Ax}_{T'_n}$ that (exactly) leads from W_n to $\{Z_n\}$.

Proof. The proof follows along the lines of Lemma 4.1. □

Corollary 4.2. The planning problem remains EXPTIME-hard even for action systems where a finite set of (unary) predicate symbols is fixed in advance, but there is no limitation for their finite domains.

Corollary 4.3. Any non-deterministic Turing machine M that accepts in space n can be faithfully axiomatized within propositional well-balanced pure Horn LL theories, as well as, within well-balanced pure Horn LL theories with a fixed number of unary predicates.

Proof. Since a non-deterministic Turing machine M can be conceived of as an alternating machine whose states may be labelled only by $\vee$, we apply just the same construction of Theorem 4.1, or its modification in Theorem 4.2.

The simplification is that we use here only pure Horn axioms of the form (13), or of the form (16). □

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

Corollary 4.4. (PSPACE-hardness)

(a) There is a `polytime` sequence of propositional LL theories T_n (each $\mathtt{Ax}_{T_n}$ consists of $O(n)$ well-balanced pure Horn axioms of $\otimes$-rank 2) such that the problem
   — of deciding, for any n and for any sequent of the Horn form $W \vdash Z$ such that both W and Z are of $\otimes$-rank $n+2$, whether this sequent is AL-derivable within T_n
   is `PSPACE`-hard.

(b) Given a finite domain, there is a `polytime` sequence of action systems $\mathscr{A}_n$ over this fixed domain (each of $\mathscr{A}_n$ consists of $O(n)$ well-balanced deterministic actions) such that the planning problem
   — of deciding, for any n and for any initial situation W and any partial goal Z, whether there exists a plan $\mathscr{P}$ based on the actions from $\mathscr{A}_n$, which leads from W to $\{Z\}$
   is `PSPACE`-hard.

Proof. The proof follows by the definition of `PSPACE`-hardness, based on Lemma 4.1. □

5. Upper bounds

5.1. *Descriptive complexity of theories*

The total number of non-equivalent elementary products is greatly affected by commutativity of $\otimes$.

Lemma 5.1. Given a theory T over a finite signature Σ:

(a) let S_T be the total number of closed atomic formulas in signature Σ, and

(b) let $L_T(m)$ denote the total number of non-equivalent closed elementary products in signature Σ whose $\otimes$-rank is bounded by m.

Then

(a) $S_T \leqslant (J \cdot D^a)$, where:
   (a1) J is the total number of predicate symbols in signature Σ,
   (a2) a is an upper bound of their arity, and
   (a3) D is the total number of constants in signature Σ.

(c) For any m:

$$L_T(m) \leqslant \binom{m + S_T}{m} = \frac{(m + S_T)!}{m! \cdot (S_T)!} \leqslant \min\{(S_T + 1)^m, (m + 1)^{S_T}\}.$$

Comment 5.1. If we do not assume commutativity of $\otimes$, the inequality $L_T(m) \leqslant (S_T + 1)^m$ still remains valid, whereas $L_T(m) \leqslant (m + 1)^{S_T}$ fails (*cf.* Comment 5.4).

To analyze computational complexity of the algorithm of Corollary 3.1, we have to introduce *two* kinds of numerical parameters:

(i) one is related to the *length of description* of theories T themselves, and

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

(ii) the other is related to the *length of description* of the *planning* tasks:
'from W to $\{Z_1, .., Z_k\}$'.

To this end, predicate formulas and sequents that may be used in LL theories are naturally encoded by means of finite binary strings.

The length of a binary string x is denoted by $|x|$.

Let Q_1, Q_2, .., Q_j, ... be an infinite list of the predicate symbols used, and let d_1, d_2, .., d_i, ... be an infinite list of the constants used.

To obtain a complete description of an elementary proposition $Q_j(d_{i_1}, d_{i_2}, .., d_{i_{k_j}})$, it is sufficient to have the following types of information: (the binary codes of) the integers j, k_j, and i_1,i_2,..,i_{k_j}. Hence, the *'binary length'* of such a proposition is

$$O(\log_2 j + k_j \log_2 \max\{i_1, i_2, .., i_{k_j}\}),$$

which amounts to $O(\log_2 S_T)$ for a given theory T with S_T from Lemma 5.1.
As a corollary,

(a) the length of the binary code of an elementary product X whose $\otimes$-rank is m is

$$O(m \cdot \log_2 S_T),$$

and

(b) the length of the binary code of the Horn clause $X \vdash (Y_1 \oplus \cdots \oplus Y_\ell)$ is

$$O(\ell \cdot m \cdot \log_2 S_T),$$

where m is the maximum of $\otimes$-ranks of X, Y_1, .., Y_ℓ.

Intuitively, the *descriptive complexity* of a given robot system is the actual size of an English description of its theory T that allows us to recognize its axioms (notwithstanding the fact that the set $\mathtt{Ax}_T$ of axioms of the robot system might be infinite).

Formally, the *honest* descriptive size of the theory T is determined by the length of the shortest program p describing the process of recognizing its axioms within the framework of a certain programming method β.

For example, the simplest method of describing theory $\mathtt{Hanoi}_N$ from Example 2.1 is to list all its axioms, which takes $\Omega(N^2)$ space.

On the other hand, all these $\Omega(N^2)$ axioms can be completely recognized in polynomial time, based only on two elements of information:

(i) a general form of sequents (5)–(8) (which is constant), and
(ii) the binary code of the number N (which is variable).

The effect is that the *honest* descriptive complexity of $\mathtt{Hanoi}_N$ does not exceed

$$\text{Const.} + \log_2 N.$$

According to Kolmogorov (Kolmogorov 1965), a *programming method* is defined as a Turing machine β with two input tapes. Binary strings p placed on one of the input tapes are conceived of as *programs*, with their *inputs* x being placed on the other input tape. By definition, $p(x) := \beta(p, x)$.

We say that a *program* p *describes* a theory T in `polytime` (`pspace`, `exptime`) within the framework of a given programming method β if for any binary string x:

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

(a) $\beta(p, x)$ runs in polynomial time (polynomial space, exponential time) with respect to $|x|$, and
(b) $\beta(p, x)=1$ if and only if x is the binary code of an axiom from $\mathtt{Ax}_T$.

Theorem 5.1. (Kolmogorov 1965) We can design a 'universal' programming method ψ so that for any programming method β there is a constant C_β such that each of the programs p describing a theory T in `polytime` (`pspace`, `exptime`) within β, can be *effectively* transformed into a program p', describing the same T in `polytime` (`pspace`, `exptime`) within this ψ, so that

$$|p'| \leqslant |p|+C_\beta.$$

Henceforth, we fix such an *asymptotically optimal* programming method ψ and measure Kolmogorov descriptive complexity within the framework of the ψ fixed.

5.2. *Upper bounds*

In order to simplify the form of our complexity results, we assume here and henceforth that in the LL theories under consideration every proper axiom of the form (11) is a *two-branching* Horn clause: $\ell \leqslant 2$.

Corollary 3.1 yields a hyper-exponential algorithm, since the size of the tree-like plan examined there is $O(2^{L_T(m)})$.

But the actual number of the values $\mathtt{value}_v(W)$ in a plan is $O(L_T(m))$, which allows us to fold the tree-like plan 'exponential times' by 'gluing' equivalent vertices (see Figure 2).

Furthermore, these folded plans can be kept overlapped, so that *the time cost of the algorithm can be greatly reduced*, as in the following theorem.

Theorem 5.2 (EXPTIME). There is an algorithm α applicable to the following pair of inputs:

(i) any recursive branching Horn LL theory T over a finite signature such that each of the axioms of T is a non-lengthening two-branching Horn clause, and
(ii) any closed elementary products W, $Z_1,..,Z_k$,

which:

(a) determines whether there is a plan based on $\mathtt{Ax}_T$ leading from W to the set of partial goals $\{Z_1,..,Z_k\}$, and
(b) if the answer is positive, makes such a plan (of the minimal height but in a folded form), and
(c) in addition to the algorithm from Corollary 3.1, algorithm α runs in time

$$O((L_T(m))^5 \times \mathtt{TIME}_\rho(4m\cdot\log_2 S_T)), \tag{18}$$

where m is the least upper bound of $\otimes$-ranks of W, $Z_1,..,Z_k$, and ρ is a program that decides the set of proper axioms $\mathtt{Ax}_T$.

(We assume that $k \leqslant L_T(m)$.)

Proof. Let $\mathscr{Z}_i$ be the set of all non-equivalent closed elementary products H of $\otimes$-rank bounded by m that lead to the desired partial goals in i steps or less.
More precisely,

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

(i) $\mathscr{Z}_0$ consists of all H of the form $(Z_j \otimes V)$, where $1 \leqslant j \leqslant k$ and V is an elementary product such that the $\otimes$-rank of $(Z_j \otimes V)$ is bounded by m.

(ii) $\mathscr{Z}_{i+1} := \mathscr{Z}_i \cup \widehat{\mathscr{Z}}_{i+1}$,
where $\widehat{\mathscr{Z}}_{i+1}$ is the set of all non-equivalent H whose $\otimes$-rank is bounded by m such that there is an *instance* of an axiom from $\mathtt{Ax}_T$ of the form $X \vdash (Y_1 \oplus Y_2)$, and two products H_1 and H_2 from $\mathscr{Z}_i$ for which the following holds:
there is a V such that $H \simeq (X \otimes V)$, $H_1 \simeq (Y_1 \otimes V)$ and $H_2 \simeq (Y_2 \otimes V)$.

By construction, for all H taken from any $\mathscr{Z}_i$, a sequent of the form

$$H \vdash (Z_1 \oplus \cdots \oplus Z_k)$$

is AL-provable within T.
In the opposite direction, for any sequent

$$W \vdash (Z_1 \oplus \cdots \oplus Z_k)$$

that is AL-provable within T, Lemma 3.1 and Theorem 3.3 yields a finite plan $\mathscr{P}$ that exactly leads from W to $\mathscr{Z}_0$, and, in addition,

(a) in each of the axioms involved inside $\mathscr{P}$ the $\otimes$-rank of each of its terms is bounded by m, and

(b) the height of $\mathscr{P}$ is bounded by $L_T(m)$.

The effect is that $W \simeq H$ for some H from $\mathscr{Z}_{L_T(m)}$.

Thus, to resolve the *decision problem* in question, it suffices to check whether $H \in \mathscr{Z}_i$ for each of the $L_T(m)$ products H, with i ranging up to $L_T(m)$:

(i) $\mathscr{Z}_0$ is produced in $k \times L_T(m)$ steps, which amounts to $(L_T(m))^2$ steps.

(ii) As for $\mathscr{Z}_{i+1}$, such a checking procedure involves only *closed instances* of axioms of the form $X \vdash (Y_1 \oplus Y_2)$, where the $\otimes$-rank of X, Y_1, Y_2 is bounded by m.
The number of 'candidates' for such an instance is $(L_T(m))^3$, and the binary length of each of them is $(3+O(1)) \cdot m \cdot \log_2 S_T$, which is less than $4m \cdot \log_2 S_T$.
Hence, we can generate the instances of axioms we need by means of the program ρ, with the input strings being of length $4m \cdot \log_2 S_T$.
(The reason we do not assume that a finite list of *all* closed instances of the proper axioms is given *in advance* is that the total number of closed instances is generally *exponential* compared to the cardinal number of $\mathtt{Ax}_T$. Furthermore, $\mathtt{Ax}_T$ might have been *infinite*.)

Bringing together all those numbers, the total time is at most

$$O(L_T(m) \times L_T(m) \times ((L_T(m))^3 \times \mathtt{TIME}_\rho(4m \cdot \log_2 S_T))).$$

Having just succeeded in getting W within some $\mathscr{Z}_t$, since we can store the axioms applied, we make the corresponding plan (in a folded form) for the same cost (18). It is worthwhile observing that we get a plan with the *minimal* 'height' t. □

Comment 5.2. For theories T described by reasonable (for instance, pspace) programs ρ, Lemma 5.1 shows that our improved algorithm requires at most an *exponential time*:

$$O((S_T + 1)^{5m} \times \mathtt{TIME}_\rho(4m \cdot \log_2 S_T)).$$

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

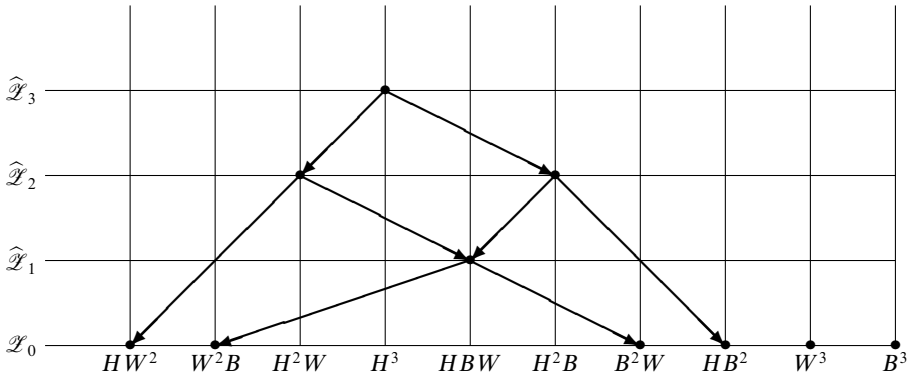


Fig. 3. How to construct the folded plan shown in Figure 2.

Figure 3 shows how to resolve the planning problem from Example 2.2 for $N=3$, $n=2$. Ten elementary products of $\otimes$-rank 3 are placed there in some order. We use $\bullet$ to indicate that the corresponding product has been included in $\widehat{\mathscr{Z}}_i$.

Since here $L_T(N)$ is $O(N^3)$, the *sock problem* is sorted out in `polytime` by the technique of Theorem 5.2.

Corollary 5.1. (PTIME for a fixed signature) Let Σ be a fixed finite signature (consisting of a finite number of predicate symbols with their arity and a finite number of constants). Then there is an algorithm β_Σ applicable to the following pair of inputs:

(i) any recursive branching Horn LL theory T with signature Σ such that each of the axioms of T is a non-lengthening two-branching Horn clause, and $\mathtt{Ax}_T$ is decidable in `polytime` by a program ρ, and

(ii) any closed elementary products $W, Z_1,..,Z_k$,

which:

(a) determines whether there is a plan based on $\mathtt{Ax}_T$, leading from W to the set of partial goals $\{Z_1,..,Z_k\}$, and,

(b) if the answer is positive, makes such a plan (of the minimal height but in a folded form), and

(c) runs in `polynomial time` with respect to the two parameters:

(c1) the binary length of a description of the program ρ, and

(c2) m, the least upper bound of $\otimes$-ranks of $W, Z_1,..,Z_k$.

Proof. By Lemma 5.1, $L_T(m) \leqslant (m+1)^{S_T}$, with S_T being fixed, which makes the total time (18) polynomial. Note that we assume k is bounded by $L_T(m)$. □

Comment 5.3. Indeed, for a fixed theory T recognizable in `polytime`, Corollary 5.1 yields a `polytime` algorithm that resolves, for any closed $W, Z_1,..,Z_k$, the planning problem of leading from W to the set of partial goals $\{Z_1,..,Z_k\}$.

Comment 5.4. If we do not assume commutativity of $\otimes$, Corollary 5.1 fails (see Comment 5.1). Furthermore, with non-commutative $\otimes$ we can get `PSPACE`-hardness even for a fixed pure Horn LL theory T over a finite signature.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

For pure Horn LL theories, we will reduce the `exptime` upper bound to the `pspace` one.

Theorem 5.3. (PSPACE) There is an algorithm α' applicable to the following pair of inputs:

(i) any pure Horn LL theory T over a finite signature such that each of the axioms of T is a non-lengthening two-branching Horn clause and $\mathtt{Ax}_T$ is decidable by a program ρ in `pspace`, and
(ii) any closed elementary products $W, Z_1,..,Z_k$,

which:

(a) determines whether there is a plan based on $\mathtt{Ax}_T$, leading from W to the set of partial goals $\{Z_1,..,Z_k\}$, and
(b) runs in `polynomial space` with respect to the two parameters:
  (b1) the binary length of a description of the program ρ, and
  (b2) $m{\cdot}\log_2 S_T$, where m is the least upper bound of $\otimes$-ranks of $W, Z_1,..,Z_k$.

(We assume that $k \leqslant L_T(m)$.)

Proof. The main ideas are as follows.

First, we show that the *decision problem* under consideration belongs to `PSPACE` by means of the following `npspace` construction.

We begin with $W_0 := W$.

For any t, if W_t has not occurred in $\mathscr{Z}_0$ yet, we guess W_{t+1} whose $\otimes$-rank is bounded by m, so that there is a plan consisting of one edge labelled by an instance $X_t \vdash Y_t$ of a Horn clause taken from $\mathtt{Ax}_T$ that leads from W_t to $\{W_{t+1}\}$. As a result, the old W_t is replaced by W_{t+1}.

Each t-th step can be performed in space $O(m{\cdot}\log_2 S_T) + \mathtt{SPACE}_\rho(3m{\cdot}\log_2 S_T)$.

By Theorems 3.1 and 3.2, $W \vdash Z$ is derivable within T iff there is a sequence of at most $L_T(m)$ guesses that ends in $\mathscr{Z}_0$.

The technique of Savitch's theorem (Aho *et al.* 1976) shows how to perform this procedure in `pspace`.

To generate the plan that is hidden inside this construction, we modify it so that, given a k, the corresponding $X_k \vdash Y_k$ will be eventually kept on additional $O(m{\cdot}\log_2 S_T)$ space. □

6. Concluding remarks

We have introduced Horn linear logic as a comprehensive logical system capable of 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, if it started with a certain initial situation. This simple fragment of linear logic provides ordinary users with an easy way of specifying their robot systems in their own terms. In particular, our choice of the minimal set of connectives $\otimes$ and $\oplus$ is in full accordance with the naive AI semantics.

Since any linear logic derivation is valid within traditional systems, the *positive* solution could be found by traditional means. Ironically, a clear advantage of our Horn linear

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

logic approach can be illustrated by *negative* situations where we have to show that there are no solutions to the planning problem.

The fact that Horn linear logic is complete with respect to the planning problems should be taken into account in the design of efficient problem-oriented logical tools.

We have established exact complexity bounds for the robot systems with only pure deterministic actions, as well as for the robot systems that allow actions with non-deterministic effects. We put great emphasis on making plans, rather than on pure decidability. In contrast with typical LL papers on complexity (Scedrov 1993), we estimate the complexity of computing the 'values' (plans), which may be exponentially large, rather than short 1 or 0 as in decision procedures.

Acknowledgements

We are greatly indebted to the referee for his insightful comments and fruitful recommendations, which have allowed us to improve the exposition of this paper.

References

Aho, A. V., Hopcroft, J. E. and Ullman, J. D. (1976) *The Design and Analysis of Computer Algorithms*, Addison-Wesley.

Bibel, W. (1986) A deductive Solution for Plan Generation. *New Generation Computing* **4** 115–132.

Chandra, A., Kozen, D. and Stockmeyer, L. (1981) Alternation. *J. of ACM* **28** (1) 114–133.

Chapman, D. (1987) Planning for Conjunctive Goals. *Artificial Intelligence* **32** 333–377.

McDermott, D. and Hendler, J. (1995) Planning: What it is, What it could be, An introduction to the Special Issue on Planning and Scheduling. *Artificial Intelligence* **76** 1–16.

Gehlot, V. and Gunter, C. A. (1990) Normal process representatives. In: *Proc. 5th Annual IEEE Symposium on Logic in Computer Science, Philadelphia* (June).

Girard, J.-Y. (1987) Linear logic. *Theoretical Computer Science* **50** (1) 1–102.

Girard, J.-Y. (1995) Linear logic: its syntax and semantics. In: Girard, J.-Y., Lafont, Y. and Regnier, L. (eds.) Advances in Linear Logic. *London Mathematical Society Lecture Notes* **222**, Cambridge University Press 1–42.

Kanovich, M. I. (1992) Horn Programming in Linear Logic is NP-complete. In: *Proc. 7th Annual IEEE Symposium on Logic in Computer Science, Santa Cruz* (June) 200–210.

Kanovich, M. I. (1994a) Linear logic as a logic of computations. *Annals of Pure and Applied Logic* **67** 183–212.

Kanovich, M. I. (1994b) The complexity of Horn Fragments of Linear Logic. *Annals of Pure and Applied Logic* **69** 195–241.

Kanovich, M. I. (1995) The Direct Simulation of Minsky machines in Linear logic. In: Girard, J.-Y., Lafont, Y. and Regnier, L. (eds.) Advances in Linear Logic. *London Mathematical Society Lecture Notes* **222**, Cambridge University Press 123–145.

Kolmogorov, A. N. (1965) Three Approaches to the Concept of the Amount of Information. *Probl. Inf. Transm.* **1** (1).

McCarthy, J. (1986) Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence* **28** 89–116.

Masseron, M., Tollu, C. and Vauzeilles, J. (1993) Generating Plans in Linear Logic: I. Actions as proofs. *Theoretical Computer Science,* **113** 349–370.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press

Nilsson, N. J. (1980) *Principles of Artificial Intelligence*, Springer.
Scedrov, A. (1993) Linear logic and computation: a survey. In: Schwichtenberg, H. (ed.) *Proof and Computation, Proc. Marktoberdorf Summer School 1993*, NATO Advanced Science Institutes, Berlin 281–298.

https://doi.org/10.1017/S0960129501003413 Published online by Cambridge University Press