Paper V

Reasoning about Action in Polynomial Time

Thomas Drakengren and Marcus Bjäreland

ABSTRACT

Although many formalisms for reasoning about action exist, surprisingly few approaches have taken computational complexity into consideration. The contributions of this paper are the following: a temporal logic with a restriction for which deciding satisfiability is tractable, a tractable extension for reasoning about action, and NP-completeness results for the unrestricted problems. Many interesting reasoning problems can be modelled, involving nondeterminism, concurrency and memory of actions. The reasoning process is proved to be sound and complete.

1 Introduction

Although many formalisms for reasoning about action exist, surprisingly few approaches have taken computational complexity into consideration. One explanation for this might be that many interesting AI problems are (at least) NP-hard, and that tractable subproblems that are easily extracted, tend to lack expressiveness. This has led a large part of the AI community to rely on heuristics and incomplete systems to solve the problems (see, for example, [Ginsberg, 1996] for a discussion). This holds, in particular, for the area of reasoning about action, where the very expressive logical formalisms provide difficult obstacles when it comes to efficient implementation.

We feel, however, that the tractability boundary for sound and complete reasoning about action has not yet been satisfactorily investigated. We prove this by introducing a nontrivial subset of a logic with semantics closely related to the *trajectory semantics* of Sandewall [1994], for which satisfiability is tractable. Our logic can handle examples involving not only nondeterminism, but continuous time, concurrency and memory of actions as well, thus providing a conceptual extension of Sandewall's framework. The reader should note that our main concern is *computation*, as opposed to *modelling*.

This paper is organised as follows: Section 2 is an informal overview of the technical results, where two examples are also presented. In Section 3 we present the syntax and semantics of the basic temporal logic and the extension for reasoning about action, and in Section 4, we present the following: three intractability results for these formalisms, and the main results: the tractable subclasses of the temporal logic and its extension.

2 Overview

In Section 3 we develop a temporal logic, Λ, which is syntactically related to the propositional temporal logic TPTL [Alur and Henzinger, 1989]. The temporal domain is the set of real numbers and temporal expressions are based on relations $=$, $\leq$, $<$, $\geq$ and $>$ between linear polynomials with rational coefficients over a set of temporal variables. The semantics of this temporal logic is standard. The formalism for reasoning about action is *narrative* based, which means that *scenario descriptions* are used to model the real world. Scenario descriptions consist of formulae in the temporal logic (*observations*) and *action expressions* which are constructs that state that certain changes in values of the *features* (propositions, fluents) may occur.

We write action expressions as $\pi \Rrightarrow [\alpha]\epsilon$ `Infl`, where π is the precondition for the action, ϵ the effects, α a temporal expression denoting *when* the effects are taking place, and `Infl` is the set of all features that are influenced by the action. The influenced features are not subject to the assumption of inertia, i.e. we allow them, and only them, to change during the execution of the action.

It turns out that deciding satisfiability is NP-complete, both for the temporal logic and the scenario descriptions. Interestingly, the problem is NP-complete for scenario descriptions that only include Horn clause observations, unconditional and unary action expressions (this terminology is explained later), and no stated relations between temporal expressions.

To extract a tractable subset from our formalism we rely on a recent result in *temporal constraint reasoning* by Jonsson and Bäckström [1996] (also discovered independently by Koubarakis [1996]). They have identified a large tractable class of temporal constraint reasoning, using *Horn Disjunctive Linear Relations* (Horn DLRs) which are relations between linear polynomials with rational coefficients. We make use of their result by restricting formulae in our scenario descriptions to be Horn and then by encoding scenario descriptions into Horn DLRs. For the temporal logic this is fairly straightforward. For the scenario descriptions, it turns out that we have to put some constraints on the temporal relations and actions in the scenario descriptions.

We will use the following two examples: Jump into a Lake with a Hat [Giunchiglia and Lifschitz, 1995] and Soup Bowl Lifting [Gelfond *et al.*, 1991]. Below we informally describe the examples.

Example 1 (Jump into a Lake with a Hat, JLH) If you jump into the lake you will get wet. If you have been in the water at some time point it is unclear if you still have your hat on. This is an example of *nondeterminism* and of *memory of actions.* □

Example 2 (Soup Bowl Lifting, SBL) If we lift either side of a soup bowl at some time points, the content will be spilled, unless we lift both sides at the same time point. This is an example of *concurrency.* □

The first example can be handled by the tractable subset of our formalism, whereas the second one fails to satisfy the restrictions. Note that actions may partly overlap.

3 Scenario Descriptions

We introduce a semantics that is a simpler variant of Sandewall's Features and Fluents Framework [Sandewall, 1994], in that the effects of an action can only occur at one and the same time point for a given action, and we use only propositional values of features (similar to the work of Doherty [1994]) However, in some respects this formalism is more flexible than Sandewall's: we use a *continuous* time domain, we allow *concurrently* executing actions, and effects of actions can depend on other states in the history than the state at the *starting time point* of the action (this implies *memory* of actions, in Sandewall's [1994] terminology). One example of a formalism having memory is that of Gustafsson and Doherty [1996].

Initially, a basic temporal logic is defined. The computational properties of this logic will be exploited by the scenario description logic, i.e. ultimately (in Section 4) the scenario descriptions will be transformed into formulae of the basic temporal logic.

3.1 Syntax

We begin by defining the basic temporal logic.

We assume that we have a set $\mathcal{T}$ of time point variables intended to take real values, and a set $\mathcal{F}$ of *features* intended to take propositional values.

Definition 3 A *signature* is a tuple $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$, where $\mathcal{T}$ is a finite set of time point variables and $\mathcal{F}$ is a finite set of propositional features. A *time point expression* is a linear polynomial over $\mathcal{T}$ with rational coefficients. We denote the set of time point expressions over $\mathcal{T}$ by $\mathcal{T}^*$. □

Definition 4 Let $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$ be a signature, let $\alpha, \beta \in \mathcal{T}^*$, $f \in \mathcal{F}$, $R \in \{=, \leq, <, \geq, >\}$, $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, and define the *scenario description language* Λ over σ by

$$\Lambda ::= \mathtt{T} \mid \mathtt{F} \mid f \mid \alpha R \beta \mid \neg\Lambda \mid \Lambda_1 \oplus \Lambda_2 \mid [\alpha]\Lambda.$$

A formula of the form $\alpha R \beta$ is a *linear relation*, and one that does not contain any connectives (any of the constructs $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$ and $[\cdot]$) is *atomic*. If γ is atomic and $\alpha \in \mathcal{T}^*$, then the formulae γ, $[\alpha]\gamma$, $\neg\gamma$, $[\alpha]\neg\gamma$ and $\neg[\alpha]\gamma$ are *literals*[1]. (A formula $[\alpha]\gamma$ expresses that at time α, γ is true.) A literal l is *negative* iff it contains $\neg$ and its corresponding atomic formula γ is not of

[1] So, the formula $\neg[\alpha]\neg\gamma$ is not a literal.

the form $\alpha R \beta$ for $R \in \{<, \leq, >, \geq\}$. A literal that is not negative is *positive*. Disjunctions of literals are *clauses*. A formula $\gamma \in \Lambda$ is in *conjunctive normal form*, CNF, iff it is a conjunction of clauses. A formula γ is *Horn* iff it is a clause with at most one positive literal. A set Γ of formulae is *Horn* iff every $\gamma \in \Gamma$ is Horn. Syntactical identity between formulae is written $\equiv$, and when ambiguity is to be avoided, we denote formulae $\gamma \in \Lambda$ by $\ulcorner \gamma \urcorner$.

Let γ be a formula. A feature $f \in \mathcal{F}$ occurs *free* in γ iff it does not occur within the scope of a $[\alpha]$ expression in γ. $\alpha \in \mathcal{T}^*$ *binds* f in γ if a formula $[\alpha]\phi$ occurs as a subformula of γ, and f is free in ϕ. If no feature occurs free in γ, γ is *closed*. If γ does not contain any occurrence of $[\alpha]$ for any $\alpha \in \mathcal{T}^*$, then γ is *propositional*. $\Box$

Using Λ, we can thus express propositions being true at time points, and express relations between time points. Next we define the extension of the basic temporal logic by introducing *action expressions*, i.e. constructs that enable modelling of change.

Definition 5 Let $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$ be a signature. An *action expression* over σ is a tuple $A = \langle \alpha, \pi, \mathtt{Infl}, \epsilon \rangle$, $\alpha \in \mathcal{T}^*$, π a closed formula in Λ, $\mathtt{Infl} \subseteq \mathcal{F}$, and ϵ a propositional formula, where all features occurring in ϵ are in $\mathtt{Infl}$. α is the *result time point* of A, π is the *precondition* of A, $\mathtt{Infl}$ is the set of *influenced features* of A, and ϵ is the *effects* of A. A is *unconditional* iff $\pi \equiv \mathtt{T}$, and *unary* iff $|\mathtt{Infl}| = 1$.

For convenience we write action expressions as

$$\pi \Rrightarrow [\alpha]\epsilon\, \mathtt{Infl};$$

for example we have

$$[3]\mathit{loaded} \Rrightarrow [4]\neg \mathit{alive}\{\mathit{alive}\}$$

for an action `shoot`. If $\pi \equiv \mathtt{T}$, we remove it and the $\Rrightarrow$ symbol, and if $\epsilon \equiv \mathtt{T}$, we remove it. An example is an unconditional loading action

$$[2]\mathit{loaded}\{\mathit{loaded}\},$$

and the action of spinning the chamber of a gun

$$[3]\{\mathit{loaded}\}.$$

An *observation* over σ is a closed formula in Λ. $\Box$

Next, we combine the concepts defined so far into one.

Definition 6 A *scenario description* is a tuple

$$\Upsilon = \langle \sigma, \text{SCD}, \text{OBS} \rangle,$$

where $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$ is a signature, SCD (the *schedule*) is a finite set of action expressions over σ, and OBS is a finite set of observations over σ. The *size* of a scenario description is defined as the sum of lengths of all formulae in SCD and OBS. □

Now, we formalise the examples from Section 2.

Example 7 (JLH) The intended conclusion of the following scenario is that the person is wet at time $\mathbf{c}_1$, and we do not know if the hat is on at time point $\mathbf{c}_2$, occurring after the person jumps.

OBS1	$[0]hat_on \wedge dry \wedge on_land$
SCD1	$[\mathbf{c}_1]\neg on_land\{on_land\}$
SCD2	$[\mathbf{c}_1]\neg on_land \Rrightarrow [\mathbf{c}_1]\neg dry\{dry\}$
SCD3	$[\mathbf{c}_1]\neg on_land \Rrightarrow [\mathbf{c}_2]\{hat_on\}$
OBS4	$\mathbf{c}_1 \geq 0 \wedge \mathbf{c}_2 \geq \mathbf{c}_1$

□

Example 8 (SBL) We have two actions: one for lifting the left side of the soup bowl and one for lifting the right side. If the actions are not executed simultaneously, the table cloth will no longer be dry. The intended conclusion here is $\mathbf{c}_2 = \mathbf{c}_1$.

OBS1	$[0]dry$
SCD1	$[\mathbf{c}_1]leftup\{leftup\}$
SCD2	$[\mathbf{c}_1]\neg rightup \Rrightarrow [\mathbf{c}_1]\neg dry\{dry\}$
SCD3	$[\mathbf{c}_2]rightup\{rightup\}$
SCD4	$[\mathbf{c}_2]\neg leftup \Rrightarrow [\mathbf{c}_2]\neg dry\{dry\}$
OBS2	$[\mathbf{c}_2]dry$
OBS3	$\mathbf{c}_2 > 0 \wedge \mathbf{c}_1 > 0$

□

3.2 Semantics

For the presentation of the semantics we proceed similarly to the presentation of the syntax. We begin by defining the semantics of the basic temporal logic.

Definition 9 Let $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$ be a signature. A *state* over σ is a function from $\mathcal{F}$ to the set $\{\mathtt{T}, \mathtt{F}\}$ of truth values. A *history* over σ is a function h from $\mathbb{R}$ to the set of states. A *valuation* ϕ is a function from $\mathcal{T}$ to $\mathbb{R}$. It is extended in a natural way (as a homomorphism from $\mathcal{T}^*$ to $\mathbb{R}$), giving e.g.

$$\phi(3t + 4.3) = 3\phi(t) + 4.3.$$

A *development*, or *interpretation*, over σ is a tuple $\langle h, \phi \rangle$ where h is a history and ϕ is a valuation. □

Definition 10 Let $\gamma \in \Lambda$, and let $D = \langle h, \phi \rangle$ be a development. Define the *truth value* of γ in D for a time point $t \in \mathbb{R}$, denoted $D(\gamma, t)$, as follows (here we overload $\mathtt{T}$ and $\mathtt{F}$ to denote both formulae and truth values). Assume $f \in \mathcal{F}$, $R \in \{=, \leq, <, \geq, >\}$, $\alpha, \beta \in \mathcal{T}^*$, $\gamma, \delta \in \Lambda$, $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, and $\tau \in \{\mathtt{T}, \mathtt{F}\}$. Now define

$$\begin{array}{lcl} D(\tau, t) & = & \tau \\ D(f, t) & = & h(t)(f) \\ D(\alpha R \beta, t) & = & \phi(\alpha) R \phi(\beta) \\ D(\neg\gamma, t) & = & \neg D(\gamma, t) \\ D(\gamma \oplus \delta, t) & = & D(\gamma, t) \oplus D(\delta, t) \\ D([\alpha]\gamma, t) & = & D(\gamma, \phi(\alpha)). \end{array}$$

Two formulae γ_1 and γ_2 are *equivalent* iff $D(\gamma_1, t) = D(\gamma_2, t)$ for all D and t. A set $\Gamma \subseteq \Lambda$ of formulae is *satisfiable* iff there exists a development D and a time point $t \in \mathbb{R}$ such that $D(\gamma, t)$ is true for every $\gamma \in \Gamma$. A development D is a *model* of a set $\Gamma \subseteq \Lambda$ of closed formulae iff $D(\gamma, t)$ is true for every $t \in \mathbb{R}$ and $\gamma \in \Gamma$. □

Fact 11 For $\gamma \in \Lambda$ and $\alpha \in \mathcal{T}^*$, $\neg[\alpha]\gamma$ is equivalent to $[\alpha]\neg\gamma$. For a closed formula γ, $D(\gamma, t) = D(\gamma, t')$ for any $t, t' \in \mathbb{R}$ and development D. □

Thus, if γ is closed, we can write $D(\gamma)$ instead of $D(\gamma, t)$.

Now we define the semantics of the action expressions based on models for the basic temporal logic. Inertia (the frame problem) is handled by identifying all time points where a feature f possibly can change its value. Then during every interval where no such change time point exists, f has to have the same value throughout the interval.

Definition 12 Let $D = \langle h, \phi \rangle$ be a development. An action expression $A = \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle$ *has effects* in D iff $D(\pi) = \texttt{T}$. Let $f \in \mathcal{F}$, and define $Chg(D, \text{SCD}, f, t)$ to be true for a time point $t \in \mathbb{R}$ iff $f \in \texttt{Infl}$ for some action expression $A = \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle \in$ SCD that has effects in D, with $\phi(\alpha) = t$. Note that we can only have $Chg(D, \text{SCD}, f, t)$ true for a finite number of time points for fixed SCD and f.

Let $\Upsilon = \langle \sigma, \text{SCD}, \text{OBS} \rangle$ be a scenario description. An *intended model* of Υ is a development $D = \langle h, \phi \rangle$ where

- D is a model of OBS
- For each $A = \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle \in$ SCD that has effects in D, $D(\epsilon, \phi(\alpha)) = \texttt{T}$
- For each $f \in \mathcal{F}$ and $s, t \in \mathbb{R}$ with $s < t$ such that for no $t' \in (s, t)$ (open interval), $Chg(D, \text{SCD}, f, t')$ holds, we have $h(t')(f) = h(s)(f)$ for every $t' \in (s, t)$.

Denote by $Mod(\Upsilon)$ the set of all intended models for a scenario description Υ.

A formula $\gamma \in \Lambda$ is *entailed* by a scenario description Υ, denoted $\Upsilon \models \gamma$, iff γ is true in all intended models of Υ. Υ is *satisfiable* iff $Mod(\Upsilon) \neq \emptyset$. □

Fact 13 If $\Upsilon = \langle \sigma, \text{SCD}, \text{OBS} \rangle$ is a scenario description and $\gamma \in \Lambda$ a formula, then $\Upsilon \models \gamma$ iff $\langle \sigma, \text{SCD}, \text{OBS} \cup \{\neg\gamma\} \rangle$ is unsatisfiable. □

We comment how this is used in our two examples:

- It is intended that JLH is unsatisfiable adding the observation $[\mathbf{c}_1]dry$, and that neither adding $[\mathbf{c}_2]hat_on$ nor $[\mathbf{c}_2]\neg hat_on$ will make it unsatisfiable.
- For SBL, adding $\mathbf{c}_2 \neq \mathbf{c}_1$ as an observation will make the scenario unsatisfiable.

4 Complexity Results

4.1 Basic Results

It is no surprise that deciding satisfiability for the basic temporal logic is NP-hard. Proofs of NP-completeness, on the other hand, depend on the tractability results.

Proposition 14 Deciding satisfiability of a set $\Gamma \subseteq \Lambda$ is NP-hard.
Proof: Propositional logic is a subset of Λ. □

Corollary 15 Deciding whether a scenario description is satisfiable is NP-hard. □

That these problems are in NP, and thus are NP-complete, is proved in Theorem 23 and Theorem 24.

Interestingly, we can strengthen the result considerably.

Theorem 16 Deciding whether a scenario description is satisfiable is NP-hard, even if action expressions are unconditional and unary, only Horn observations are allowed, and no disjunctive relations between time points may be stated.
Proof: Reduction from 3SAT. Let $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ be the clauses of a 3CNF formula ϕ, and v the set of propositional symbols used in ϕ. We construct a scenario description Υ satisfying the required restrictions as follows. Set

$$\mathcal{F} = v \cup \{p' \mid p \in v\},$$

$$\mathcal{T} = \{s, t\} \cup \{t_p \mid p \in v\},$$

$$\text{SCD} = \{\langle t, \mathtt{T}, \{p\}, \neg p\rangle \mid p \in v\} \cup \{\langle t, \mathtt{T}, \{p'\}, p'\rangle \mid p \in v\},$$

$$\begin{aligned}\text{OBS} = \ & \{t > s\} \cup \{[s]p, [s]\neg p' \mid p \in v\} \cup \\ & \{\mathcal{C}(l_{i1}) \vee \mathcal{C}(l_{i2}) \vee \mathcal{C}(l_{i3})\},\end{aligned}$$

where $\mathcal{C}(p) = \neg[t_p]p'$ and $\mathcal{C}(\neg p) = \neg[t_p]p$.

We shall use intended models D as models for ϕ, interpreting $D(t_p < t)$ as the truth value of p, and so show that Υ is satisfiable iff ϕ is satisfiable. First note the facts that in any $D \in \mathit{Mod}(\Upsilon)$, $D(\neg[t_p]p') = D([t_p]p)$, and that $D([t_p]p) = D(t_p < t)$.
$\Rightarrow$) Suppose $D \in \mathit{Mod}(\Upsilon)$, and consider a clause $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ in ϕ. By the construction of OBS, one of $\mathcal{C}(l_{i1})$, $\mathcal{C}(l_{i2})$ or $\mathcal{C}(l_{i3})$ has to be true in D, say $l = l_{i1}$. If $l = p$, then $\mathcal{C}(l) = \neg[t_p]p'$; thus $D([t_p]p)$ is true, and so is $D(t_p < t)$. If $l = \neg p$, then $\mathcal{C}(l) = \neg[t_p]p$; thus $D([t_p]p)$ is false, and so is $D(t_p < t)$.
$\Leftarrow$) Suppose ϕ has a propositional model M, and consider a clause $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ in ϕ. Construct an intended model D of Υ by letting features have values as forced by the scenario, t having the value 0 and s the value -1, and for each $p \in v$, if p is true in M, then set $t_p = -1$, and otherwise $t_p = 1$. It is clear that the expression $\mathcal{C}(l_{i1}) \vee \mathcal{C}(l_{i2}) \vee \mathcal{C}(l_{i3})$ is true in D. The result follows, since it is clear that the reduction is polynomial. □

We now present the key to tractability, which is a linear-programming approach to temporal constraint reasoning, by Jonsson and Bäckström [1996].

Definition 17 Let α and β be linear polynomials with rational coefficients over some set X of variables. Then a *disjunctive linear relation*, DLR, is a disjunction of one or more expressions of the form $\alpha = \beta$, $\alpha \neq \beta$, $\alpha \leq \beta$, $\alpha < \beta$. A DLR is *Horn* iff it contains at most one disjunct with the relation $=$, $<$ or $\leq$.

An assigment m of variables in X to real numbers is a *model* of a set Γ of DLRs iff all formulae in Γ are true when taking the values of variables in the DLRs. A set of DLRs is *satisfiable* iff it has a model. □

The following result is the main result of Jonsson and Bäckström [1996].

Proposition 18 Deciding satisfiability of a set of Horn DLRs is polynomial. □

Now we restrict the scenario description language and the form of actions. Furthermore, a structural restriction on scenario descriptions, verifiable in polynomial time, is imposed. We shall define an encoding function that takes a Horn scenario description Υ and returns a set Γ of Horn DLRs such that Γ is satisfiable iff Υ is satisfiable.

4.2 Satisfiability of Horn Formulae is Tractable

First, we code Horn formulae as Horn DLRs.

Definition 19 Let l be a closed literal, and assume the existence of fresh, unique time point variables t_f^α for each $f \in \mathcal{F}$ and $\alpha \in \mathcal{T}^*$. Then $C(l)$ is defined as follows, assuming $R \in \{=, \leq, <, \geq, >\}$ and $f \in \mathcal{F}$.

$$
\begin{array}{lcl}
C(\mathtt{T}) & = & \ulcorner 0 = 0 \urcorner \\
C(\mathtt{F}) & = & \ulcorner 0 \neq 0 \urcorner \\
C(\alpha R \beta) & = & \ulcorner \alpha R \beta \urcorner \\
C(\neg\alpha = \beta) & = & \ulcorner \alpha \neq \beta \urcorner \\
C(\neg\alpha \leq \beta) & = & \ulcorner \alpha > \beta \urcorner \\
C(\neg\alpha < \beta) & = & \ulcorner \alpha \geq \beta \urcorner \\
C(\neg\alpha \geq \beta) & = & \ulcorner \alpha < \beta \urcorner \\
C(\neg\alpha > \beta) & = & \ulcorner \alpha \leq \beta \urcorner \\
C([\eta]\alpha R \beta) & = & C(\alpha R \beta) \\
C([\alpha] f) & = & \ulcorner t_f^\alpha = 0 \urcorner \\
C(\neg[\alpha]\gamma) & = & C([\alpha]\neg\gamma) \\
C([\alpha]\neg f) & = & \ulcorner t_f^\alpha \neq 0 \urcorner
\end{array}
$$

Let $\gamma \in \Lambda$ be a closed Horn formula, and let γ' be obtained from γ by simplifying away occurrences of $\mathtt{T}$ and $\mathtt{F}$. Now γ' is on the form $\bigvee_i l_i$. Then define $C(\gamma)$ to be the DLR $\delta = \bigvee_i C(l_i)$. Note that δ is always a Horn DLR.

Let $\Gamma \subseteq \Lambda$ be a set of closed Horn formulae, and T the set of all time point expressions occurring in Γ. Then $C(\Gamma)$ is defined by

$$C(\Gamma) = \{ C(\gamma) \mid \gamma \in \Gamma \} \cup \{ C(\neg[\alpha]f \vee \beta \neq \alpha \vee [\beta]f) \mid f \in \mathcal{F}, \alpha, \beta \in T \}.$$

The second set is called the *correspondence equations*. Note that the argument of C in a correspondence equation is equivalent to

$$[\alpha]f \wedge \beta = \alpha \rightarrow [\beta]f.$$

□

The following result is crucial.

Theorem 20 Let $\Gamma \subseteq \Lambda$ be a set of closed Horn formulae. Then Γ is satisfiable iff $C(\Gamma)$ is satisfiable.
Proof: Let T be the set of time point expressions occurring in Γ.
$\Rightarrow$) Let $D = \langle h, \phi \rangle$ be a model of Γ. We shall construct a model m of $C(\Gamma)$. First set $m(t) = \phi(t)$ for all $t \in \mathcal{T}$. Now all temporal relations from Γ are satisfied in m, since they are directly transferred. For each $f \in \mathcal{F}$ and $\alpha \in T$, if $h(\phi(\alpha))(f)$ is true, then set $m(t_f^\alpha) = 0$, otherwise set $m(t_f^\alpha) = 1$. It is clear that the correspondence equations are satisfied by this definition, and so are the remaining elements of $C(\Gamma)$.
$\Leftarrow$) Let m be a model of $C(\Gamma)$, Construct an interpretation $D = \langle h, \phi \rangle$ as follows. First set $\phi(t) = m(t)$ for all $t \in \mathcal{T}$. It is enough to determine h for values $\phi(\alpha)$, $\alpha \in T$, since we have no restrictions on other values. Set $h(\phi(\alpha))(f)$ to be true iff $m(t_f^\alpha) = 0$. That h is well-defined follows directly from the correspondence equations, which hold in m. □

Corollary 21 Deciding satisfiability of sets of closed Horn formulae is polynomial.
Proof: It is clear that the transformation C is polynomial. The result follows from Proposition 18. □

4.3 Satisfiability is in NP

Now we have the results for the proofs of membership in NP for the satisfiability problems of Λ and of scenario descriptions. First, we need an auxiliary definition.

Definition 22 Let $\Gamma \subseteq \Lambda$. Then we define $\neg\Gamma = \{\neg\gamma \mid \gamma \in \Gamma\}$, $\mathit{feat}(\Gamma)$ to be the set of all features occurring in Γ, $\mathit{linrel}(\Gamma)$ to be the set of all linear relations (which are atomic formulae) occurring in Γ, and $\mathit{time}(\Gamma)$ to be the set of all time point expressions occurring in Γ. $\Box$

Theorem 23 Deciding satisfiability of a set $\Gamma \subseteq \Lambda$ is NP-complete.
Proof: By Proposition 14, it remains to prove that the problem is in NP. Let $\Gamma \subseteq \Lambda$ be a set of formulae. A few auxiliary definitions will be needed for the proof.

Let $F = \mathit{feat}(\Gamma)$, $L = \mathit{linrel}(\Gamma)$, $T = \mathit{time}(\Gamma)$, let $\mathtt{t}$ be a fresh time point variable, set

$$A^+ = \{[\alpha]f \mid \alpha \in T \cup \{\mathtt{t}\} \wedge f \in F\} \cup L,$$

and

$$A' = A^+ \cup \neg A^+.$$

We say that a set $W \subseteq A'$ is a *syntactic* Λ*-interpretation* iff for any $\phi \in A^+$, exactly one of ϕ and $\neg\phi$ is a member of W.

Next, we define a way to evaluate a formula $\gamma \in \Lambda$ in a syntactic Λ-interpretation, following Definition 10. Let $\gamma \in \Lambda$. We define the *truth value* of γ in W for $\alpha' \in \mathcal{T}^*$, denoted $W(\gamma, \alpha')$, as follows. Assume $f \in \mathcal{F}$, $R \in \{=, \leq, <, \geq, >\}$, $\alpha, \beta \in \mathcal{T}^*$, $\gamma, \delta \in \Lambda$, and $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

- $W(\mathtt{T}, \alpha') = \mathtt{T}$
- $W(\mathtt{F}, \alpha') = \mathtt{F}$
- $W(f, \alpha') = (\ulcorner[\alpha']f\urcorner \in W)$
- $W(\alpha R\beta, \alpha') = (\ulcorner\alpha R\beta\urcorner \in W)$
- $W(\neg\gamma, \alpha') = \neg W(\gamma, \alpha')$
- $W(\gamma \oplus \delta, \alpha') = W(\gamma, \alpha') \oplus W(\delta, \alpha')$
- $W([\alpha]\gamma, \alpha') = W(\gamma, \alpha)$

It is clear that $W(\gamma, \alpha')$ is always defined for $\gamma \in \Lambda$. Thus we can say that a syntactic Λ-interpretation W is a *syntactic* Λ*-model* of a set Γ of formulae iff W is satisfiable, and for all $\gamma \in \Gamma$, $W(\gamma, \mathtt{t})$ is true.

If we can prove that whenever Γ is satisfiable, then there exists a syntactic Λ-model W of Γ, and vice versa, then NP-membership will follow, since

- the size of W is polynomial in the size of Γ
- it can be checked in polynomial time whether W is a syntactic Λ-interpretation or not
- $W(\gamma, \alpha')$ can easily be computed in polynomial time
- by Corollary 21, checking satisfiability of a set of closed Horn formulae is polynomial, and thus checking satisfiability of W is polynomial.

$\Rightarrow$) Suppose that Γ is satisfiable, i.e. that for some $t \in \mathbb{R}$, $I(\gamma, t) = \mathtt{T}$ for all $\gamma \in \Gamma$. Construct a syntactic Λ-interpretation W from $I = \langle h, \phi \rangle$ by first setting $I' = \langle h, \phi' \rangle$, where $\phi'(s) = \phi(s)$ for all $s \in \mathcal{T} - \{\mathtt{t}\}$, and $\phi'(\mathtt{t}) = t$, and then defining

$$W = \{\delta \in A' \mid I'(\delta) = \mathtt{T}\}.$$

It remains to check that W is satisfiable, and that $W(\gamma, \mathtt{t}) = \mathtt{T}$ for all $\gamma \in \Gamma$.

W is trivially satisfiable, since by construction I' is a model of W. We prove by induction on γ that $W(\gamma, \alpha') = I(\gamma, \phi'(\alpha'))$ for all $\gamma \in \Lambda$ and $\alpha' \in T \cup \{\mathtt{t}\}$; thus $W(\gamma, \mathtt{t}) = \mathtt{T}$ for all $\gamma \in \Gamma$.

First, the basis cases: If $\gamma \equiv \mathtt{T}$ or $\gamma \equiv \mathtt{F}$, the result is immediate. If $\gamma \equiv f$ for $f \in \mathcal{F}$, then

$$\begin{aligned} & W(\gamma, \alpha') = \\ & \quad = W(f, \alpha') \\ & \quad = (\ulcorner [\alpha'] f \urcorner \in W) \\ & \quad = I'([\alpha'] f) \\ & \quad = I(f, \phi'(\alpha')). \end{aligned}$$

If $\gamma \equiv \alpha R \beta$ for $\alpha, \beta \in \mathcal{T}^*$ and $R \in \{=, \leq, <, \geq, >\}$, then

$$\begin{aligned} & W(\gamma, \alpha') = \\ & \quad = W(\alpha R \beta, \alpha') \\ & \quad = (\ulcorner \alpha R \beta \urcorner \in W) \\ & \quad = I'(\alpha R \beta, \phi'(\alpha')) \end{aligned}$$

Now, the induction: If $\gamma \equiv \neg\delta$, then

$$\begin{aligned} & W(\gamma, \alpha') = \\ & \quad = W(\neg\delta, \alpha') \\ & \quad = \neg W(\delta, \alpha') \\ & \quad = \neg I(\delta, \phi'(\alpha')) \quad \text{induction} \\ & \quad = I(\neg\delta, \phi'(\alpha')). \end{aligned}$$

If $\gamma \equiv \delta_1 \oplus \delta_2$ for $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then

$$\begin{aligned} & W(\gamma, \alpha') = \\ & \quad = W(\delta_1 \oplus \delta_2, \alpha') \\ & \quad = W(\delta_1, \alpha') \oplus W(\delta_2, \alpha') \\ & \quad = I(\delta_1, \phi'(\alpha')) \oplus I(\delta_2, \phi'(\alpha')) \quad \text{induction} \\ & \quad = I(\delta_1 \oplus \delta_2, \phi'(\alpha')). \end{aligned}$$

If $\gamma \equiv [\alpha]\delta$, then

$$\begin{aligned} & W(\gamma, \alpha') = \\ & \quad = W([\alpha]\delta, \alpha') \\ & \quad = W(\delta, \alpha) \\ & \quad = I(\delta, \phi'(\alpha)) \qquad \text{induction} \\ & \quad = I([\alpha]\delta, \phi'(\alpha')). \end{aligned}$$

$\Leftarrow$) Suppose that W is a syntactic Λ-model of Γ, i.e. that W is a satisfiable syntactic Λ-interpretation, and $W(\gamma, \mathtt{t}) = \mathtt{T}$ for all $\gamma \in \Gamma$. Let $I = \langle h, \phi \rangle$ be a model of W (note that W contains only closed formulae). We need to show that $I(\gamma, \phi(\mathtt{t})) = \mathtt{T}$ for all $\gamma \in \Gamma$. By induction we prove the stronger result that $I(\gamma, \phi(\alpha')) = W(\gamma, \alpha')$ for all $\gamma \in \Lambda$ and $\alpha' \in T \cup \{\mathtt{t}\}$.

First, the basis cases: If $\gamma \equiv \mathtt{T}$ or $\gamma \equiv \mathtt{F}$, the result is immediate. If $\gamma \equiv f$ for $f \in \mathcal{F}$, then

$$\begin{aligned} & I(\gamma, \phi(\alpha')) = \\ & \quad = I(f, \phi(\alpha')) \\ & \quad = I([\alpha']f) \\ & \quad = (\ulcorner[\alpha']f\urcorner \in W) \\ & \quad = W(f, \alpha'). \end{aligned}$$

If $\gamma \equiv \alpha R \beta$ for $\alpha, \beta \in \mathcal{T}^*$ and $R \in \{=, \leq, <, \geq, >\}$, then

$$\begin{aligned} & I(\gamma, \phi(\alpha')) = \\ & \quad = I(\alpha R \beta, \phi(\alpha')) \\ & \quad = (\alpha R \beta \in W) \\ & \quad = W(\alpha R \beta, \alpha'). \end{aligned}$$

Now, the induction: If $\gamma \equiv \neg\delta$, then

$$\begin{aligned} & I(\gamma, \phi(\alpha')) = \\ & \quad = I(\neg\delta, \phi(\alpha')) \\ & \quad = \neg I(\delta, \phi(\alpha')) \\ & \quad = \neg W(\delta, \alpha') \\ & \quad = W(\neg\delta, \alpha'). \end{aligned}$$

If $\gamma \equiv \delta_1 \oplus \delta_2$ for $\oplus \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then

$$\begin{aligned} & I(\gamma, \phi(\alpha')) = \\ & \quad = I(\delta_1 \oplus \delta_2, \phi(\alpha')) \\ & \quad = I(\delta_1, \phi(\alpha')) \oplus I(\delta_2, \phi(\alpha')) \\ & \quad = W(\delta_1, \alpha') \oplus W(\delta_2, \alpha') \\ & \quad = W(\delta_1 \oplus \delta_2, \alpha'). \end{aligned}$$

If $\gamma \equiv [\alpha]\delta$, then

$$\begin{aligned} & I(\gamma, \phi(\alpha')) = \\ & \quad = I([\alpha]\delta, \phi(\alpha') \\ & \quad = I(\delta, \phi(\alpha)) \\ & \quad = W(\delta, \alpha) \\ & \quad = W([\alpha]\delta, \alpha'). \end{aligned}$$

Thus the result follows. □

Theorem 24 Deciding whether a scenario description is satisfiable is NP-complete.

Proof: By Corollary 15, it remains to prove that the problem is in NP. Let $\Upsilon = \langle \sigma, \text{SCD}, \text{OBS} \rangle$ be a scenario description. A few auxiliary definitions will be needed for the proof.

Define the set Δ of formulae by

$$\Delta = \text{OBS} \cup \{\ulcorner \pi \rightarrow [\alpha]\epsilon \urcorner \mid \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle \in \text{SCD}\},$$

let $T = \mathit{time}(\Delta)$, $F = \mathit{feat}(\Delta)$, $L = \mathit{linrel}(\Delta)$, and define

$$\Delta' = \{\alpha R \beta \mid \alpha, \beta \in T \wedge R \in \{=, <, >\}\},$$

$$A^+ = \{[\alpha]f \mid \alpha \in T \wedge f \in F\} \cup L \cup \Delta',$$

and

$$A' = A^+ \cup \neg A^+.$$

We say that a set $W \subseteq A'$ is a *syntactic scenario interpretation* iff for any $\phi \in A^+$, exactly one of ϕ and $\neg\phi$ is a member of W.

By employing exactly the same method as in Theorem 23, we can define the *truth value* in W of a formula $\gamma \in \Delta \cup \Delta'$, denoted $W(\gamma)$ (no temporal parameter is needed, since all formulae in $\Delta \cup \Delta'$ are closed). Exactly as in Theorem 23, W satisfies the following:

- If I is a model of $\Delta \cup \Delta'$, and we set

$$W = \{\delta \in A' \mid I(\delta) = \mathtt{T}\},$$

  then W is a syntactic Λ-model of $\Delta \cup \Delta'$, and I is a model of W.

- If W is a syntactic Λ-model of $\Delta \cup \Delta'$ which is satisfiable by an interpretation I, then I is a model of $\Delta \cup \Delta'$.

Two more auxiliary definitions are needed, for W being a syntactic scenario interpretation.

First, let $\alpha, \beta \in W$. Then we say that α *precedes* β in W, written $\alpha \lhd \beta$, iff $\alpha < \beta \in W$, and for no $\beta' \in T$, $\alpha < \beta' \in W$ and $\beta' < \beta \in W$ holds. If for no α', $\alpha' \lhd \beta$, we write $-\infty \lhd \beta$, and if for no β', $\alpha \lhd \beta'$, we write $\alpha \lhd \infty$.

Second, W is a *syntactic scenario model* of the scenario description Υ iff the following holds:

- W is a syntactic Λ-model of Δ

- For each $f \in \mathcal{F}$ and $\alpha, \beta \in T$ such that $\alpha \lhd \beta$ in W, if there is no action expression $\langle \beta, \pi, \mathtt{Infl}, \epsilon \rangle \in$ SCD with $f \in \mathtt{Infl}$ for some π, $\mathtt{Infl}$ and ϵ, then $[\alpha]f \in W$ iff $[\beta]f \in W$.

Now, the following remains in order to prove the result of the theorem:

- If Υ is satisfiable, then there exists a syntactic scenario model W for Υ

- If there exist a syntactic scenario model W for Υ, then Υ is satisfiable.

This would suffice for proving NP-membership, due to the following facts:

- The size of W is polynomial in Υ

- It can be checked in polynomial time whether W is a syntactic Λ-model of $\Delta \cup \Delta'$ or not by the same argument as in Theorem 23

- Checking the second condition of the definition of syntactic scenario model is polynomial, since we can start by sorting elements of T by the order imposed on T by W (α comes before β iff $\alpha < \beta \in W$), and then proceed with the simple checks of SCD, which is obviously polynomial.

$\Rightarrow$) Suppose that Υ is satisfiable with an intended model $D = \langle h, \phi \rangle$. Now D is a clearly a model of Δ, and we can set

$$W = \{\delta \in A' \mid D(\delta) = \mathtt{T}\}.$$

Clearly W is a syntactic Λ-model of Δ. It remains to check the second condition in the definition of syntactic scenario model. Take $f \in \mathcal{F}$ and $\alpha, \beta \in T$ satisfying the required conditions, and suppose there is no action expression $\langle \beta, \pi, \mathtt{Infl}, \epsilon \rangle \in \text{SCD}$ such that $f \in \mathtt{Infl}$. By the definition of T and our assumption, we cannot have that $\mathit{Chg}(D, \text{SCD}, f, t')$ is true for any t' with $D(\alpha) < t' \leq D(\beta)$. Since there are only finitely many points where Chg is true, we can choose a time point $t'' > D(\beta)$ such that for no $s \in [D(\beta), t'')$, $\mathit{Chg}(D, \text{SCD}, f, s)$ is true. Now, the definition of intended model yields that $[\alpha]f$ is true iff $[\beta]f$ in D, and by the definition of W, that $[\alpha]f \in W$ iff $[\beta]f \in W$.

$\Leftarrow$) Suppose that W is a syntactic scenario model for Υ, which is satisfied by a model $I = \langle h, \phi \rangle$. We construct an intended model $D = \langle h', \phi \rangle$ from I as follows. For each feature $f \in F$ do the following. First for each α such that $-\infty \lhd \alpha$ in W, set $h'(t)(f) = h(\phi(\alpha))(f)$ for every $t \leq \phi(\alpha)$. Then for each α, β with $\alpha \lhd \beta$ in W, set $h'(t)(f) = h(\phi(\alpha))(f)$ for every t such that $\phi(\alpha) \leq t < \phi(\beta)$. Finally, for each α such that $\alpha \lhd \infty$, set $h'(t)(f) = h(\phi(\alpha))(f)$ for every $t \geq \phi(\alpha)$. It is clear that h' is defined for every f and t. It remains to verify that D is an intended model for Υ.

Since W is a syntactic scenario model for Υ, it is also a syntactic Λ-model of Δ. By the fact that I is a model of W, and since D and I agree on the values of all time point expressions in T, D is also a model of W, and by what we know about syntactic Λ-interpretations, D is a model of Δ. Thus, what is left to verify is the second condition in the definition of intended model.

For this purpose, let f be a feature and $s, t \in \mathbb{R}$ with $s < t$, such that for no $t' \in (s, t)$, $\mathit{Chg}(D, \text{SCD}, f, t')$ holds. We want to prove that $h'(t')(f) = h'(s)(f)$ for every $t' \in (s, t)$. Suppose to the contrary that for some t' with $s < t' < t$, $h'(t')(f) \neq h'(s)(f)$. It is easy to see that by the construction of D from I, this cannot hold, and we have a contradiction. Thus D is an intended model of Υ. $\square$

4.4 Tractable Scenario Descriptions

Using Corollary 21, we see that if we can code scenario descriptions into sets of Horn formulae, we will have a polynomial algorithm for reasoning with

scenario descriptions. In order to obtain such a result, we need to restrict what scenario descriptions are allowed.

The strategy can briefly be described as follows: we identify all observation time points which bind a feature value and all time points where an action expression possibly can change a feature value. Then we connect bound literals with biconditionals, between time points where the literal value should not change. For example, if some action expression changes the value of the feature f at time point α, there exists a $\gamma \in \text{OBS}$ which binds f at a time point β, $\alpha < \beta$, and no changes of the value of f occurs between α and β, then $[\alpha]f \leftrightarrow [\beta]f$ should be added to the theory. This formula can be rewritten in Horn form. The example represents one of the six cases (case 3). The other cases are similar.

The restrictions are basically two: First we will have to represent action expressions as Horn formulae (*restricted* action expressions). Second, the scenario descriptions must be *ordered*; for example, we could not remove the restriction $\alpha < \beta$ in the example above.

Definition 25 Let $\Upsilon = \langle \sigma, \text{SCD}, \text{OBS} \rangle$ be a scenario description. For each $f \in \mathcal{F}$, define

$$E_f = \{\alpha \mid \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle \in \text{SCD} \wedge f \in \texttt{Infl}\}$$

and

$$C_f = \{\alpha \mid \alpha \textit{ binds } f \textit{ in } \gamma \wedge \gamma \in O\},$$

for

$$O = \text{OBS} \cup \{\pi \mid \langle \alpha', \pi, \texttt{Infl}, \epsilon \rangle \in \text{SCD}\}.$$

E_f is *ordered* iff for $\alpha, \beta \in E_f$, exactly one of $\alpha < \beta$, $\alpha = \beta$ and $\alpha > \beta$ is consistent with[2] OBS. For $\alpha, \beta \in E_f$, E_f ordered, we define $\alpha \prec_f \beta$ iff $\alpha < \beta$ is consistent with OBS and for every $\gamma \in E_f$, $\alpha < \gamma < \beta$ is inconsistent with OBS, $-\infty \prec_f \alpha$ iff for no $\beta \in E_f$, $\beta < \alpha$ is consistent with OBS, and $\alpha \prec_f \infty$ iff for no $\beta \in E_f$, $\alpha < \beta$ is consistent with OBS.

Let $\alpha \in E_f$, $\beta \in \mathcal{T}^*$, and define $\alpha \ll_f \beta$ iff $\alpha \leq \beta$ is consistent with OBS, $\alpha > \beta$ is inconsistent with OBS, and for every $\gamma \in E_f$, $\alpha < \gamma \leq \beta$ is inconsistent with OBS. Also define $-\infty \ll_f \beta$ iff for every $\alpha \in E_f$ $\alpha \leq \beta$ is inconsistent with OBS.

[2] A formula γ is consistent with a set Γ iff $\Gamma \cup \{\gamma\}$ is satisfiable.

If for all $f \in \mathcal{F}$, E_f is ordered, and for all $\omega \in C_f$, $\alpha \ll_f \omega$ for some $\alpha \in E_f \cup \{-\infty\}$, then Υ is *ordered*. The last condition says that for each observation of a feature f, there is a unique change of f which sets its value, or it precedes all changes of f. □

Proposition 26 Testing whether a scenario description Υ is ordered is polynomial, if OBS is Horn.
Proof: For the orderedness of E_f, check for each feature that all pairs of result time points satisfy the condition. Since OBS is Horn, Corollary 21 guarantees that this can be done in polynomial time. The check for $\ll_f$ is polynomial in the same way. □

Definition 27 Let $A = \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle$ be an action expression. Then A is *restricted* iff either of the following holds:

- π is `T` and ϵ is a conjunction of propositional Horn formulae
- π is a disjunction of negative literals, and ϵ is either `T`, or a conjunction of negative literals.

An ordered scenario description $\Upsilon = \langle \sigma, \text{SCD}, \text{OBS} \rangle$ is *restricted* iff every action expression in SCD is restricted and OBS is Horn. A restricted scenario description is *normal* iff it is ordered, and for every action expression $A \in$ SCD either of the following holds:

- π is `T` and ϵ is a propositional Horn formula
- π is a negative literal and ϵ is either `T` or a negative literal.

□

Of the two examples previously stated, Example 7 is restricted (breaking apart the conjunction of the first observation into separate formulae), whereas Example 8) is not, which is easy to verify (the second one fails to be ordered).

The following result will make the forthcoming proofs easier.

Proposition 28 Let Υ be a restricted scenario description. Then we can construct an equivalent normal scenario description Υ' in polynomial time.
Proof: We note that a restricted action expression

$$A = \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle \in \text{SCD}$$

where $\pi \equiv \bigvee_i \neg l_i$ is equivalent to replacing it by one action expression $\langle \alpha, \neg l_i, \texttt{Infl}, \epsilon \rangle$ for each disjunct in π. Similarly, a restricted action expression $A = \langle \alpha, \pi, \texttt{Infl}, \bigwedge_i \phi_i \rangle$ can be split into action expressions A_i by $A_i = \langle \alpha, \pi, \texttt{Infl}, \phi_i \rangle$ (or even easier if $\epsilon \equiv \texttt{T}$), and obtaining the same set of intended models. The transformation is clearly polynomial.

Orderedness is preserved, since we do not change the order in which features are changed. □

Thus, we can assume that our restricted scenario descriptions are normal. Next, we define the function Φ which transforms scenario descriptions into sets of Horn formulae.

Definition 29 First let $A = \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle$ be a normal action expression. We have three cases for Φ:

- If $\pi \equiv \texttt{T}$ and $\epsilon \equiv \bigvee_j l_j$ is propositional Horn, then define

$$\Phi(A) = \{\bigvee_j [\alpha] l_j\}$$

- If $\pi \equiv \neg l$ and $\epsilon \equiv \texttt{T}$, then define

$$\Phi(A) = \emptyset$$

- If $\pi \equiv \neg l$ and $\epsilon \equiv \neg m$, then define

$$\Phi(A) = \{l \vee [\alpha] \neg m\}.$$

The restriction to normal action expressions should be clear; it implies that $\Phi(A)$ is Horn. For a set $\mathcal{S}$ of action expressions, define

$$\Phi(\mathcal{S}) = \bigcup_{A \in \mathcal{S}} \Phi(A).$$

Let $\Upsilon = \langle \sigma, \text{SCD}, \text{OBS} \rangle$ be a restricted scenario description with $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$, and without loss of generality, assume that each feature in $\mathcal{F}$ occurs in SCD or OBS. Also let b be a fresh time point variable (b standing for "beginning"), and for each $f \in \mathcal{F}$, add a new feature f'. A few construction steps (basically corresponding to the possible relations between time points in E_f and C_f) are necessary. We provide the intuitions behind the constructions as we present them.

1. $\Gamma_1 = \text{OBS} \cup \Phi(\text{SCD}) \cup \{b < \alpha \mid \alpha \in E_f \cup C_f, f \in \mathcal{F}\}$.

   The observations, the transformed action expressions, and an initial time point are added.

2. $\Gamma_2 = \{\neg[\alpha]f \vee [b]f, [\alpha]f \vee \neg[b]f \mid f \in \mathcal{F}, \alpha \in C_f, -\infty \ll_f \alpha\}$.

   No action expression influences f before α where it is bound by an observation, therefore f should have the same value at b as at α. Note that the members of Γ_2 each are equivalent to $[\alpha]f \leftrightarrow [b]f$.

3. $\Gamma_3 = \{\neg[\alpha]f \vee [\beta]f, [\alpha]f \vee \neg[\beta]f \mid f \in \mathcal{F}, \alpha \in E_f, \beta \in C_f, \alpha \ll_f \beta\}$.

   f is influenced at α and bound by an observation at a later time point β. No actions have effects between α and β, therefore f should have the same value at β as it had at α.

4. $\Gamma_4 = \{\neg[\alpha]f' \vee [b]f, [\alpha]f' \vee \neg[b]f \mid f \in \mathcal{F}, \alpha \in E_f, -\infty \prec_f \alpha\}$.

   This case resembles case 2, with the difference that f is influenced at α. Therefore, we introduce a new feature symbol f' which has the same value at α as f has at b. The new symbols will be handled properly below, in case 6.

5. $\Gamma_5 = \{\neg[\alpha]f \vee [\beta]f', [\alpha]f \vee \neg[\beta]f' \mid f \in \mathcal{F}, \alpha, \beta \in E_f, \alpha \prec_f \beta\}$.

   This relates to case 3, just as case 4 relates to case 2.

6. Since E_f is ordered, we can form equivalence classes T_f^α of time points for $\alpha \in E_f$ by

   $$T_f^\alpha = \{\beta \in E_f \mid \alpha = \beta \text{ is consistent with OBS}\}.$$

   For each $\alpha \in E_f$, define

   $$\mathcal{P}_f^\alpha = \{\pi \mid \langle \beta, \pi, \texttt{Infl}, \epsilon\rangle \in \text{SCD} \wedge f \in \texttt{Infl} \wedge \beta \in T_f^\alpha\}.$$

   Now set

   $$\Gamma_6 = \{\bigvee \mathcal{P}_f^\alpha \vee \neg[\alpha]f \vee [\alpha]f', \bigvee \mathcal{P}_f^\alpha \vee \neg[\alpha]f' \vee [\alpha]f \mid \alpha \in E_f\}.$$

   First note that this set is equivalent to the set

   $$\{\neg \bigvee \mathcal{P}_f^\alpha \rightarrow ([\alpha]f \leftrightarrow [\alpha]f') \mid \alpha \in E_f\}.$$

Here we ensure that when actions do not have effects, f will have the same value as it had the last time it was changed. This value is held by the feature f'.

Now finally set

$$\Phi(\Upsilon) = \bigcup_i \Gamma_i.$$

It is clear that the transformation performed by Φ is polynomial. □

Theorem 30 Let Υ be a restricted scenario description, and set $\Gamma = C(\Phi(\Upsilon))$. Then Υ is satisfiable iff Γ is satisfiable.
Proof: We start by defining a set Γ' which is satisfiable iff Γ is, by first defining sets Γ'_i for $i \in \{1, \ldots, 6\}$, giving $\Gamma' = \bigcup_{1 \leq i \leq 6} \Gamma'_i$:

1. $\Gamma'_1 = \Gamma_1$
2. $\Gamma'_2 = \{[\alpha]f \leftrightarrow [b]f \mid f \in \mathcal{F}, \alpha \in C_f, -\infty \ll_f \alpha\}$
3. $\Gamma'_3 = \{[\alpha]f \leftrightarrow [\beta]f \mid f \in \mathcal{F}, \alpha \in E_f, \beta \in C_f, \alpha \ll_f \beta\}$
4. $\Gamma'_4 = \{[\alpha]f' \leftrightarrow [b]f \mid f \in \mathcal{F}, \alpha \in E_f, -\infty \prec_f \alpha\}$
5. $\Gamma'_5 = \{[\alpha]f \leftrightarrow [\beta]f' \mid f \in \mathcal{F}, \alpha, \beta \in E_f, \alpha \prec_f \beta\}$
6. $\Gamma'_6 = \{\neg(\bigvee_i \pi_i) \rightarrow ([\alpha]f \leftrightarrow [\alpha]f') \mid \mathtt{T} \notin \{\pi_i\} = \mathcal{P}_f^\alpha, \alpha \in E_f\}$.

It is easy to see that each Γ'_i is satisfiable iff the corresponding Γ_i is.
$\Rightarrow$) Suppose Υ is satisfiable by an intended model $D = \langle h, \phi \rangle$. We first construct a new interpretation $D' = \langle h', \phi' \rangle$ as follows.

- Define ϕ' by setting $\phi'(s) = \phi(s)$ for all $s \in \mathcal{T} - \{b\}$, and $\phi'(b) = \min\{\phi(\alpha) \mid \alpha \in T\} - 1$.
- For each feature f, set $h'(t)(f) = h(t)(f)$.
- For each feature f, introduce a new feature f', and define $h'(t)(f')$ so that the formulae in Γ'_4 and Γ'_5 are all true. It is clear that this is possible, since these are essentially *definitions*.

We want to prove that D' is a model of Γ. This is done by checking for every $i \in \{1, \ldots, 6\}$, that D' is a model of Γ'_i.

1. For Γ'_1, all formulae in OBS are true in D', by definition. Furthermore, all formulae in $\Phi(\text{SCD})$ will be true, since these code exactly the conditions required by an intended model, in terms of actions. Furthermore, we have set $\phi'(b)$ to be strictly smaller than the value of every time point expression used, so everything in the last part of Γ'_1 will be true in D'.

2. Γ'_2 says that for each observation or action precondition involving a feature f that is not preceded by any effect on the feature f, the value will be the same as the value at b. Since the value of b in D' is less than every time point expression used, this amounts to strict inertia in each f at all time points before the first change in f. This is clearly satisfied in D', since it inherits this property from the intended model D.

3. Γ'_3 says that for any feature f, observations and action precondition involving f coming directly after changes of f with no change in between, f should retain its value from the change. This is the same as inertia, which is clearly satisfied in any intended model, and is inherited from D.

4. Γ'_4 is satisfied by construction.

5. Γ'_5 is satisfied by construction.

6. Γ'_6 says that if none of the actions which can cause a change in f at α has a true precondition, then f and f' coincide at $\alpha \in E_f$. Since by the definitions of the f' formulae, if $\beta \in E_f$, then f' is always forced to have the value at β that f had just before it changed at β; this just amounts to inertia, which holds for f by construction.

$\Leftarrow$) Suppose Γ is satisfiable by a model $I = \langle h, \phi \rangle$. We construct an intended model $D = \langle h', \phi \rangle$ of Υ from I as follows. Start by removing all features f' for features f. Then for each feature f, do the following. First, for each $\alpha \in E_f$ such that $-\infty \prec_f \alpha$, set $h'(t)(f) = h(\phi(b))(f)$ for each t such that $t < \phi(\alpha)$. Then for each $\alpha, \beta \in E_f$ such that $\alpha \prec_f \beta$, set $h'(t)(f) = h(\phi(\alpha))(f)$ for each t such that $\phi(\alpha) \leq t < \phi(\beta)$. Finally, for each $\alpha \in E_f$ such that $\alpha \prec_f \infty$, set $h'(t)(f) = h(\phi(\alpha))(f)$ for each $t \geq \phi(\alpha)$. It is clear that h' is defined for every t and f, and that by construction, for each f and $\alpha \in E_f$, $h(\phi(\alpha))(f) = h'(\phi(\alpha))(f)$. It remains to verify that D is an intended model of Υ.

First note that D is still a model of $\Gamma'_1 \cup \Gamma'_2 \cup \Gamma'_3$, since the truth of all these formulae is preserved by the transformation from I to D. Thus, D is a model of OBS, since all values used to evaluate the truth of formulae in OBS are identical in I and D, due to D being a model of Γ'_2 and Γ'_3.

For the second condition of the definition of intended model, note that the set

$$\Delta = \{\ulcorner \pi \rightarrow [\alpha]\epsilon \urcorner \mid \langle \alpha, \pi, \texttt{Infl}, \epsilon \rangle \in \text{SCD}\}$$

is satisfiable iff $\Phi(\text{SCD})$ is, by construction, and that this is equivalent to the second condition being satisfied. Thus, since I is a model of $\Phi(\text{SCD})$, I is also a model of Δ. It remains to check that D is also a model of Δ. Exactly as for the previous condition, the truth values of the π component will not change from I to D. Furthermore, by construction, D will have the same values as I on the $[\alpha]\epsilon$ expressions, since values of features f at time points $\alpha \in E_f$ are the same in D and I. Thus D is also a model of Δ.

Now the proof for the third condition. Suppose that for a feature f and time points $s, t \in \mathbb{R}$ with $s < t$, we have that for no $t' \in (s,t)$, $Chg(D, \text{SCD}, f, t')$ holds, but $h'(t')(f) \neq h'(s)(f)$ for some $t' \in (s,t)$. Now, it has to hold that $t' = \phi(\alpha')$ for some $\alpha' \in E_f$, by the construction of D.

First suppose $-\infty \prec_f \alpha'$. By the construction of D, we have that $h'(s)(f) = h'(\phi(b))(f)$, and that $h'(\phi(b))(f) = h(\phi(b))(f)$, and thus $h'(s)(f) = h(\phi(b))(f)$. Since I is a model of Γ'_4, $h(\phi(\alpha'))(f') = h'(s)(f)$; so by assumption, $h(\phi(\alpha'))(f') \neq h'(\phi(\alpha'))(f)$, and by the construction of D, it holds that $h(\phi(\alpha'))(f') \neq h(\phi(\alpha'))(f)$. I is a model of Γ'_6; thus $\neg \bigvee \mathcal{P}_f^{\alpha'}$ must be false in I, and so π is true in I for some $\langle \alpha', \pi, \texttt{Infl}, \epsilon \rangle \in \text{SCD}$ with $f \in \texttt{Infl}$. But this means that $Chg(D, \text{SCD}, f, t')$, a contradiction.

Then suppose $\alpha \prec_f \alpha'$ for some $\alpha \in E_f$. It will suffice to find a contradiction for $s = \phi(\alpha)$, so we make this assumption. Now, by the construction of D, we have that $h'(\phi(\alpha))(f) = h(\phi(\alpha))(f)$, and thus $h'(s)(f) = h(\phi(\alpha))(f)$. Since I is a model of Γ'_5, $h(\phi(\alpha'))(f') = h'(s)(f)$; so by assumption, $h(\phi(\alpha'))(f') \neq h'(\phi(\alpha'))(f)$, and by the construction of D, it holds that $h(\phi(\alpha'))(f') \neq h(\phi(\alpha'))(f)$. I is a model of Γ'_6; thus $\neg \bigvee \mathcal{P}_f^{\alpha'}$ must be false in I, and so π is true in I for some $\langle \alpha', \pi, \texttt{Infl}, \epsilon \rangle \in \text{SCD}$ with $f \in \texttt{Infl}$. But this means that $Chg(D, \text{SCD}, f, t')$, a contradiction. The result follows. □

Theorem 31 Deciding satisfiability (and entailment) for restricted scenario descriptions is polynomial. □

5 Discussion

One piece of related work is the approach by Schwalb *et al.* [1994] to reasoning about propositions being true at time points. Their choice for obtaining an algorithm is to code both propositions and temporal relations into propositional logic, whereas we do the opposite. However, their tractable inference algorithm is not complete, and they define no measure on when the correct inferences will be obtained, so it is very difficult to relate it to our approach. Furthermore they cannot handle inertia adequately: there, propositions may always change when actions are performed, but certainly this is undesirable if actions which do not affect all features are used.

In this paper our concern has been computational complexity for reasoning about action. It is important to note that although we have provided polynomial algorithms for the reasoning tasks, these can hardly be considered efficient. The important results, however, are that there *exist* polynomial algorithms; the next obvious step is to also make them *fast*. For efficient implementation, there is one direction we are particularly interested in investigating: since the technique used for achieving tractability can be described as an encoding of our logic as temporal constraints for which there is a tractable algorithm for determining satisfiability, it should be possible to do something similar for other tractable temporal algebras, for example those identified by Drakengren and Jonsson [1996, 1997]. Also, an algorithm for a purely *qualitative* scenario description language (i.e. not involving metric time) would probably have a faster satisfiability-checker.

We have shown that satisfiability of scenario descriptions is NP-complete within our formalism. We feel that it would be a mistake to interpret this negatively. On the contrary, one could argue (in lines with [Gottlob, 1996]) that this would imply that many approximations, powerful heuristics and non-trivial tractable subsets of problems for reasoning about action remain to be found. This paper is a step on the way in this endeavour.

6 Conclusions

We have presented a temporal logic and an extension for reasoning about action from which tractable subsets have been extracted. This has been done with an encoding of the logic to Horn DLRs. The formalism is narrative based with continuous time, and the world is modelled with scenario descriptions consisting of action expressions and observations. It is possible

to model nondeterminism, concurrency and memory of actions. Time is represented using linear polynomials with rational coefficients over real valued variables.

References

[AAAI '96, 1996] American Association for Artificial Intelligence. *Proceedings of the 13th (US) National Conference on Artificial Intelligence (AAAI '96)*, Portland, OR, USA, August 1996.

[Alur and Henzinger, 1989] R. Alur and T.A. Henzinger. A really temporal logic. In *Proceedings of the 30th Annual IEEE Symposium on the Foundations of Computer Science*, pages 164–169. IEEE Computer Society, 1989.

[Doherty, 1994] Patrick Doherty. Reasoning about action and change using occlusion. In Anthony G. Cohn, editor, *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI '94)*, Amsterdam, Netherlands, August 1994. Wiley.

[Doyle and Aiello, 1996] J. Doyle and L. Aiello, editors. *Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR '96)*, Cambridge, MA, USA, October 1996. Morgan Kaufmann.

[Drakengren and Jonsson, 1996] Thomas Drakengren and Peter Jonsson. Maximal tractable subclasses of Allen's interval algebra: Preliminary report. In AAAI '96 [1996], pages 389–394.

[Drakengren and Jonsson, 1997] Thomas Drakengren and Peter Jonsson. Eight maximal tractable subclasses of Allen's algebra with metric time. *Journal of Artificial Intelligence Research*, 7:25–45, 1997.

[Gelfond *et al.*, 1991] M. Gelfond, V. Lifschitz, and A. Rabinov. What are the limitations of the situation calculus? In R. Boyer, editor, *Essays in Honor of Woody Bledsoe*, pages 167–179. Kluwer Academic, 1991.

[Ginsberg, 1996] M. Ginsberg. Do computers need common sense? In Doyle and Aiello [1996].

[Giunchiglia and Lifschitz, 1995] Enrico Giunchiglia and Vladimir Lifschitz. Dependent fluents. In Chris Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 1964–1969, Montréal, PQ, Canada, August 1995. Morgan Kaufmann.

[Gottlob, 1996] G. Gottlob. Complexity and expressive power of KR formalisms. In Doyle and Aiello [1996].

[Gustafsson and Doherty, 1996] J. Gustafsson and P. Doherty. Embracing occlusion in specifying the indirect effects of actions. In Doyle and Aiello [1996].

[Jonsson and Bäckström, 1996] Peter Jonsson and Christer Bäckström. A linear-programming approach to temporal reasoning. In AAAI '96 [1996], pages 1235–1240.

[Koubarakis, 1996] M. Koubarakis. Tractable disjunctions of linear constraints. In *Proceedings of the 2nd International Conference on Principles and Practice for Constraint Programming*, pages 297–307, Cambridge, MA, August 1996.

[Sandewall, 1994] Erik Sandewall. *Features and Fluents.* Oxford University Press, 1994.

[Schwalb *et al.*, 1994] E. Schwalb, K. Kask, and R. Dechter. Temporal reasoning with constraints on fluents and events. In *Proceedings of the 12th (US) National Conference on Artificial Intelligence (AAAI '94)*, Seattle, WA, USA, July–August 1994. American Association for Artificial Intelligence.