Expressive Reasoning about Action in Nondeterministic Polynomial Time

Thomas Drakengren
Laxgatan 4, 2tr
S-133 43 Saltsjobaden, Sweden
thodrQfjuk.org

Marcus Bjareland*
Dept of Comp and Info Science
Linkoping University
S-581 83 Linkoping, Sweden
marbjOida.liu.se

Abstract

The rapid development of efficient heuristics for deciding satisfiability for propositional logic motivates thorough investigations of the usability of NP-complete problems in general. In this paper we introduce a logic of action and change which is expressive in the sense that it can represent most propositional benchmark examples in the literature, and some new examples involving parallel composition of actions, and actions that may or may not be executed. We prove that satisfiability of a scenario in this logic is NP-complete, and that it subsumes an NP-complete logic (which in turn includes a nontrivial polynomial-time fragment) previously introduced by Drakengren and Bjareland.

1 Introduction

The rapid development of efficient heuristics for deciding satisfiability for propositional logic (GSAT and similar heuristics [Selman *et a*/., 1992]) motivates thorough investigations of the usability of NP-complete problems in general. In this paper we introduce a logic of action and change which is expressive in the sense that it can represent most propositional[1] benchmark examples in the literature, and some new examples involving parallel composition of actions, and actions that may or may not be executed. We prove that this logic extends a previous formalism, where sati lability for scenarios is NP-complete, and for which reasoning could be done in polynomial time in a fragment of the logic [Drakengren and Bjareland, 1997]. Although the polynomial class they characterize is nontrivial, its expressiveness is very limited. This also holds for the tractable fragments of the action description language *A* [Gelfond and Lifschitz, 1993], found by Liberatore [1997]. In that work, Liber a-tore showed that satisfiability of *A* is NP-complete, and he provided an encoding of *A* into propositional logic in order to able to use e.g. GSAT. There seems to be no other analyses of computational complexity for similar formalisms, for instance for the Situation Calculus [McCarthy and Hayes, 1969].

*This research has been supported in parts by the Swedish Research Council for Engineering Sciences (TFR), the Knut and Alice Wallenberg Foundation, and ASF A Brown Bovery (dept ISY/AMC).

[1] This means that fluents are propositional.

2 Overview

We will develop a logic for action and change in similar spirit as in Drakengren and Bjareland [1997] (henceforth, for convenience we shall denote this paper *DB97),* i.e. we will define the syntax and semantics of a temporal propositional logic (syntactically, but not semantically related to TPTL [Alur and Henzinger, 1989]) with the possibility of expressing time points as linear polynomials with rational coefficients. Then we introduce the notion of a *scenario description,* which is basically a set of formulae in the temporal logic, and formulae that describe the change of fluent values over time. An *interpretation* of a scenario description is a function, which given a set of formulae yields the possibilities as to what *combinations of changes* are allowed by the set of formulae. For instance, a scenario description may allow either the combination of changes that the feature *loaded* and *alive* both are set to false at time 3, or the combination that *gunner Sleeps* is set to true at time 3. A *model* of a scenario description is defined as an interpretation such that for some possible combination of changes, fluents change values iff they are explicitly stated to change by this combination of changes. This way of defining interpretations enables us to (semantically) model composition of actions as operations on sets of combinations of changes, and to (syntactically) model them on the object-level. In fact, we will view every Boolean combination of changes as an object-level composition of separate actions, in an arbitrary number of levels. The basis case is then the *change features,* which are syntactical constructs that state that the value of one feature changes to true or false.

For the results regarding computation, the NP-hardness of satisfiability in our logic is obvious, since we can encode propositional logic in it. NP-membership, on the other hand, is more tricky to prove: the obvious way of guessing an interpretation and a combination of changes and verifying that the interpretation is a model using this combination of changes in polynomial time fails, since the set of combinations of changes can be exponentially large. Instead, we guess also *how* a combination of changes is obtained from the interpretation, and this solves the problem. However, this proof technique fails if we introduce quantification over time points, which is used in approaches to ramification (see e.g. Gustafsson and Doherty [1996]).

3 Scenario Descriptions

This section defines syntax and semantics of the logic together with some illustrating examples.

3.1 Syntax

We begin by defining a slight (syntactical) extension of the temporal logic in DB97. What differs is that the binary operator $\vee_x$ for *exclusive or* is included, but it is straightforward to check that all relevant results of DB97 hold with this extension.

Definition 1 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 2 Let $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$ be a signature, let $\alpha, \beta \in \mathcal{T}^*$, $f \in \mathcal{F}$, $R \in \{=, \leq, <, \geq, >\}$, $\oplus \in \{\wedge, \vee, \vee_x, \rightarrow, \leftrightarrow\}$, and define the *scenario description language* Σ over *a* by

$$\Sigma ::= \mathtt{T} \mid \mathtt{F} \mid f \mid \alpha R \beta \mid \neg\Sigma \mid \Sigma_1 \oplus \Sigma_2 \mid [\alpha]\Sigma.$$

A formula $[\alpha]\gamma$ expresses that at t i rα, γ s true. The remaining connectives are standard (the notation for exclusive or, $\vee_x$, is however nonstandard). Whenever we say "formula", we mean a formula in Σ, for a given σ.

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 *[a]* expression in γ. If no feature occurs free in γ, γ is *closed**

The *size* of a set of formulae is the sum of the lengths of the formulae in the set. □

We shall be informal with respect to specifying over what signature the language Σ is defined, when it is clear what signature is intended.

In DB97 a special language for expressing action scenarios, extending the basic temporal logic, is defined. In this paper, we shall not need that; we just modify the semantics of the temporal logic, and identify a distinguished set of features which designate *change* of another feature. This is the natural generalisation of Sandewall's concept of *occlusion,* which in turn is equivalent to what is accomplished by a *release* statement in the language $\mathcal{AR}$ [Giunc higlia *et a*l., 1997].

Definition 3 (Scenario description) Let $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$ be a signature, $\mathcal{I} \subseteq \mathcal{F}$, $\Delta = \{\delta(f, \mathtt{T}), \delta(f, \mathtt{F}) \mid f \in \mathcal{F}\}$, $\sigma' = \langle \mathcal{T}, \mathcal{F} \cup \Delta \rangle$ (that is, we consider $\delta(f, \mathtt{T})$ and $\delta(f, \mathtt{F})$ to be features themselves), and Γ a finite set of closed formulae over σ'. Then $\Upsilon = \langle \sigma, \Gamma, \mathcal{I} \rangle$ is said to be a *scenario description.* We will call features of the form $\delta(f, \mathtt{T})$ and $\delta(f, \mathtt{F})$ *change features.* Given Υ, the set Δ is denoted $\Delta(\Upsilon)$. □

The set $\mathcal{I}$ consists of all fluents that are supposed to be inert, i.e. features that will not change value over time, unless explicitly stated. The fluents not in $\mathcal{I}$ are non-inert. Intuitively, the set Δ of features is intended to model that the corresponding features in $\mathcal{I}$ can be subject to change, and remain unchanged otherwise.

We now present some examples in order to present some properties of our formalism, and to try to convince the reader that most propositional benchmark examples in the literature (see e.g. Sandewall [1994] for a list) can be represented.

Example 4 We take an example, the Russian Turkey Shoot (RTS), by Sandewall [1994], where we initially (at time point 0) assume that a turkey is alive, and that a gun is unloaded. Then we load the gun (at time point 1), spin the chamber (at 3), and fire the gun (between 4 and 5). The intended conclusion of this scenario is that we cannot prove that the turkey is alive, or dead, after the firing action has been executed. Semantically we want one model where the turkey is dead, and one where it is alive, after the execution of the firing action. We now formalise the scenario:

$$\Upsilon_{RTS} = \langle \langle \emptyset, \{alive, loaded\} \rangle, \ \Gamma_{RTS}, \{alive, loaded\} \rangle,$$

where

$$\begin{aligned} \Gamma_{RTS} = \{ & [0] alive \wedge \neg loaded, \\ & [1]\delta(loaded, \mathtt{T}), \\ & [3]\delta(loaded, \mathtt{T}) \vee \delta(loaded, \mathtt{F}), \\ & [4] loaded \rightarrow \\ & \quad [5]\delta(alive, \mathtt{F}) \wedge \delta(loaded, \mathtt{F})\}. \end{aligned}$$

□

Example 5 As an example of parallel composition we use the Balls and Boxes Scenario (BBS): We have two non-empty boxes on the ground, standing side by side. Our first action, O] is to drop a ball between the boxes, so that it is unknown in which box the ball will end up. The second action, ti2, is to throw a ball into the left box. Initially both boxes are empty. We are interested in the case when both *a* and a_2 happen. We use the features *left Empty,* and *rightEmpty,* which cire true if the left, or the right box is empty, respectively.

$$\begin{aligned} \Gamma_{Boxes} = \{ & [0] leftEmpty \wedge rightEmpty, \\ & [\mathtt{t}]\delta(leftEmpty, \mathtt{F}) \vee_x \\ & \quad \delta(rightEmpty, \mathtt{F}), \\ & [\mathtt{t}]\delta(leftEmpty, \mathtt{F})\}. \end{aligned}$$

The effect of both *a* and a_2 happening, as described by the second and third formulae in the set above, as we will see below, will be

$$\begin{aligned} & [\mathtt{t}](\delta(leftEmpty, \mathtt{F}) \vee_x \delta(rightEmpty, \mathtt{F})) \wedge \\ & \quad \delta(leftEmpty, \mathtt{F}). \end{aligned}$$

The intended conclusion is that either only the right box is empty, or both boxes are nonempty. We cannot get the second case by formulating the composite action in a straightforward manner in e.g. PMON[2] [Doherty, 1994], since

$$((\neg H(\mathbf{t}, \mathit{leftEmpty}) \vee_x \neg H(\mathbf{t}, \mathit{rightEmpty}))\wedge \\ \neg H(\mathbf{t}, \mathit{leftEmpty}))\wedge \\ \mathit{Occlude}(\mathbf{t}, \mathit{leftEmpty}) \wedge \mathit{Occlude}(\mathbf{t}, \mathit{rightEmpty})$$

is equivalent to

$$(\neg H(\mathbf{t}, \mathit{leftEmpty}) \wedge H(\mathbf{t}, \mathit{rightEmpty}))\wedge \\ \mathit{Occlude}(\mathbf{t}, \mathit{leftEmpty}) \wedge \mathit{Occlude}(\mathbf{t}, \mathit{rightEmpty}),$$

i.e. the box to the right will be empty. □

Example 6 Finally, we model actions that may or may not be executed as $[\mathbf{t}](A \vee \mathrm{T})$, where *A* is decribes an action, and T the truth value *true.* When we define the semantics of scenario descriptions, the intuition behind this syntax will be clear. Logics like PMON will have similar problems with this construct, but for reasons of space, an example of that is omitted. □

For an example with interactions between concurrently executed actions, see DB97 (which is the *soup bowl lifting* example of [Baral and Gelfond, 1997]).

3.2 Semantics

We now define the semantics of the temporal logic. The semantics will be directly defined in terms of scenario descriptions; note however that the semantics will be identical to the standard semantics of the basic temporal logic of DB97 if we set $\mathcal{I} = \emptyset$.

Definition 7 Let $\sigma = \langle \mathcal{T}, \mathcal{F} \rangle$ be a signature. A *state* over σ is a function from $\mathcal{F}$ to the set $\{\mathrm{T}, \mathrm{F}\}$ 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, giving e.g. $\phi(3t + 4.3)$ ^ $3\phi(t) + 4.3$. An *interpretation* over σ is a tuple $I = \langle h, \phi \rangle$ where h is a history and ϕ is a valuation.

Similarly for a scenario description Υ with signature σ, *1* is an interpretation for T iff / is an interpretation over σ. □

In ordinary logic, we use the notion of *truth value* for a formula in a model. Here we shall genereralise this, so that we instead obtain a set of *possible combinations of changes,* where an empty such set corresponds to F, and a nonempty set corresponds to T, but with possibly several alternatives for changes of actions (this is nondeterminism on the semantical level).

Definition 8 (Combination of changes, possible combinations of changes) Let $\Upsilon = \langle \sigma, \Gamma, \mathcal{I} \rangle$ be a scenario description, $\gamma = \bigwedge \Gamma$, and let $I = \langle h, \phi \rangle$ be an interpretation over *a.* A set *e* of tuples $\langle t, f, \tau \rangle$, where $t \in \mathbb{R}$, $f \in \mathcal{I}$ and $\tau \in \{\mathrm{T}, \mathrm{F}\}$ is said to be a *combination of changes* (one tuple therein is said to be a *change),* and a set *E* of such combinations is said to be *a set of possible combinations of changes.*

It seems that most other logics of action and change would have similar problems.

Given Γ, / and £, the set of *possible combinations of changes* of Γ in J for a time point $t \in \mathbb{R}$, denoted $I(\Gamma, t)$, is a set *E* (note that this is a set of *sets)* of possible combinations of changes is defined below. For this we first need an auxiliary function *condeff (B, E),* taking a truth value *B* and a set *E* of possible combinations of changes, returning *E* if *B* is true, and $\emptyset$ otherwise.

Let $f \in \mathcal{F}$, $R \in \{=, \leq, <, \geq, >\}$, $\alpha, \beta \in \mathcal{T}^*$, $\gamma, \epsilon \in \Sigma$, $\oplus \in \{\wedge, \vee, \vee_x, \rightarrow, \leftrightarrow\}$, and $\tau \in \{\mathrm{T}, \mathrm{F}\}$. Now define

$$\begin{array}{lcl} I(\Gamma, t) & = & I(\bigwedge \Gamma), t) \\ I(\tau, t) & = & \mathit{condeff}(\tau, \{\emptyset\}) \\ I(f, t) & = & \mathit{condeff}(h(t)(f), \{\emptyset\}) \\ I(\delta(f, \mathrm{T}), t) & = & \{\{\langle t, f, \mathrm{T} \rangle\}\} \\ I(\delta(f, \mathrm{F}), t) & = & \{\{\langle t, f, \mathrm{F} \rangle\}\} \\ I(\alpha R \beta, t) & = & \mathit{condeff}(\phi(\alpha) R \phi(\beta), \{\emptyset\}) \\ I(\neg\gamma, t) & = & \neg I(\gamma, t) \\ I(\gamma \oplus \epsilon, t) & = & I(\gamma, t) \oplus I(\epsilon, t) \\ I([\alpha]\gamma, t) & = & I(\gamma, \phi(\alpha)), \end{array}$$

where the operators $\neg$, $\wedge$, $\vee$, $\vee_x$, $\rightarrow$ and $\leftrightarrow$ on sets *E*, E_2 of possible combinations of changes as follows.

$$\begin{array}{lcl} \neg E_1 & = & \begin{cases} \emptyset & \text{if } E_1 \neq \emptyset \\ \{\emptyset\} & \text{otherwise} \end{cases} \\ E_1 \wedge E_2 & = & \{e_1 \cup e_2 \mid e_1 \in E_1, e_2 \in E_2\} \\ E_1 \vee E_2 & = & \{e_1, e_2, e_1 \cup e_2 \mid e_1 \in E_1, e_2 \in E_2\} \\ E_1 \vee_x E_2 & = & \{e_1 \in E_1 \mid \neg\exists e_2 \in E_2 . e_2 \subseteq e_1\} \cup \\ & & \{e_2 \in E_2 \mid \neg\exists e_1 \in E_1 . e_1 \subseteq e_2\} \\ E_1 \rightarrow E_2 & = & \begin{cases} E_2 & \text{if } E_1 \neq \emptyset \\ \{\emptyset\} & \text{otherwise} \end{cases} \\ E_1 \leftrightarrow E_2 & = & (E_1 \rightarrow E_2) \wedge (E_2 \rightarrow E_1) \end{array}$$

□

Since the result of / does not depend on *t* when γ is closed, we can write $I(\gamma)$ in these cases. If $I(\gamma) \neq \emptyset$, we say that γ is *true* in /.

With this definition of interpretations, it is clear that we model *any* Boolean combination of combinations of changes as composition. For example, disjunction is non-deterministic composition, and conjunction is parallel composition (with "parallel" here, we denote that the changes could *possibly* be simultaneous). However, note that negated change features (e.g. as antecedents in implication) do *not* affect the resulting set of combinations of changes. For example, the formula $[\mathbf{t}]\delta(f_1, \mathrm{T}) \rightarrow \delta(f_2, \mathrm{T})$ is equivalent to $\mathrm{T} \rightarrow \delta(f_2, \mathrm{T})$, since a negated change is interpreted to $\emptyset$ or $\{\emptyset\}$: that is, we do not model that something should *not* change. This of course means that implication is *causal* and not material (see e.g. McCain and Turner [1995]).

Intuitively, we should interpret the empty set of possible combinations of changes so that there is no possible combination of effects that could have taken place, and the set *containing* only the empty set as its possible combination of changes that the only possibility with respect to changes is that *nothing* changes. This is coherent with how the *truth value* is defined above.

We will briefly discuss how our three examples are interpreted:

- For RTS, the interesting part to interpret is the spinning action, which yields:

$$I([3]\delta(loaded, \mathrm{T}) \vee [3]\delta(loaded, \mathrm{F})) =$$
$$I([3]\delta(loaded, \mathrm{T})) \vee I([3]\delta(loaded, \mathrm{F})) =$$
$$I(\delta(loaded, \mathrm{T}), 3) \vee I(\delta(loaded, \mathrm{F}), 3) =$$
$$\{\{\langle 3, loaded, \mathrm{T}\rangle\}\} \vee \{\{\langle 3, loaded, \mathrm{F}\rangle\}\} =$$
$$\{\{\langle 3, loaded, \mathrm{T}\rangle\}, \{\langle 3, loaded, \mathrm{F}\rangle\},$$
$$\{\langle 3, loaded, \mathrm{T}\rangle, \langle 3, loaded, \mathrm{F}\rangle\}\}$$

Thus, we have three possibilities of change due to the spinning action, that *loaded* becomes true, that it becomes false, or that it becomes both true and false (this is, of course, impossible, and it is taken care of by the histories, as we will see below).

- For BBS, we focus on the composite action $a_1 \wedge a_2$, i.e.

$$[\mathrm{t}](\delta(leftEmpty, \mathrm{F}) \vee_x \delta(rightEmpty, \mathrm{F})) \wedge$$
$$\delta(leftEmpty, \mathrm{F}),$$

for which we get the following possible combinations of changes:

$$(\{\{\langle \mathrm{t}, leftEmpty, \mathrm{F}\rangle\}\} \vee_x \{\{\langle \mathrm{t}, rightEmpty, \mathrm{F}\rangle\}\}) \wedge$$
$$\{\{\langle \mathrm{t}, leftEmpty, \mathrm{F}\rangle\}\}.$$

By definition of the operator $\vee_x$ on sets of possible combinations of changes, we get the following:

$$\{\{\langle \mathrm{t}, leftEmpty, \mathrm{F}\rangle\}, \{\langle \mathrm{t}, rightEmpty, \mathrm{F}\rangle\}\} \wedge$$
$$\{\{\langle \mathrm{t}, leftEmpty, \mathrm{F}\rangle\}\} =$$
$$\{\{\langle \mathrm{t}, leftEmpty, \mathrm{F}\rangle\},$$
$$\{\langle \mathrm{t}, rightEmpty, \mathrm{F}\rangle, \langle \mathrm{t}, leftEmpty, \mathrm{F}\rangle\}\}$$

There are two possibilities: in the first combination of changes, only the right box will be empty, whereas in the second, both boxes are nonempty.

- For the "maybe" action, we note that

$$I([\mathrm{t}]A \vee [\mathrm{t}]\mathrm{T}) = I(A, \mathrm{t}) \vee I(\mathrm{T}, \mathrm{t}) = I(A, \mathrm{t}) \vee \{\emptyset\}.$$

The resulting set will include $\{\emptyset\}$, which means that it is possible that the resulting action has no effects at all, which is what we intended.

Proposition 9 Let Γ be a set of formulae over a signature σ containing no formulae of the type $\delta(f, \mathrm{T})$ or $\delta(f, \mathrm{F})$. Then / is a model of Γ in the sense of DB97 iff $I(\Upsilon) \neq \emptyset$ for the scenario description $\Upsilon = \langle \sigma, \Gamma, \emptyset\rangle$.
Proof: An easy induction shows first that $I(\gamma) \in \{\emptyset, \{\emptyset\}\}$, and then that $\emptyset$ corresponds to F and $\{\emptyset\}$ to T according to the definitions of DB97. □

Finally, we need to code the fact that actions succeed, and *inertia* (the frame problem) into the formalism. This is done by identifying all time points where a feature / can possibly change its value, exactly as in DB97. During every interval where no such change time point exists, / has to have the same value throughout the interval.

When looking at the following definition, it is instructive to look at the corresponding definition in DB97.

Definition 10 Let $\Upsilon = \langle \sigma, \Gamma, \mathcal{I}\rangle$ be a scenario description. A *mode?* of T is an interpretation $I = \langle h, \phi\rangle$ for which there exists an $e \in I(\Gamma)$ such that

- for each $f \in \mathcal{I}$ and $s, t \in \mathbb{R}$ with $s < t$ such that for no $t' \in (s, t]$ (half-open interval) $\langle t', f, \tau\rangle \in e$ holds (for some $\tau \in \{\mathrm{T}, \mathrm{F}\}$), we have $h(t)(f) = h(s)(f)$
- for each $\langle t, f, \tau\rangle \in$ e, it holds that $h(t)(f) = \tau$.

Intuitively, this definition ensures that no change in the value of a feature occurs in an interval if nothing changes it explicitly, and all specified changes have effect. Note that e is always a finite set, so this definition makes sense[4].

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

A formula $\gamma \in \Sigma$ is *entailed* by a scenario description Υ, denoted $\Upsilon \models \gamma$, iff 7 is true in all models of Υ. Υ is *satisfiable* iff $Mod(\Upsilon) \neq \emptyset$. □

Fact 11 If $\Upsilon = \langle \sigma, \Gamma, \mathcal{I}\rangle$ is a scenario description and $\gamma \in \Sigma$ a formula, then $\Upsilon \models \gamma$ **iff** $\langle \sigma, \Gamma \cup \{\neg\gamma\}, \mathcal{I}\rangle$ is unsatisfiable. □

Next we establish that this formalism indeed subsumes that of DB97. An auxiliary result is needed[5].

Definition 12 (Corresponding action formula) Let a be the action expression $\phi \Rightarrow [\alpha]\psi$Lnf1 (in the sense of DB97). Then the formula

$$\phi \rightarrow [\alpha](\psi \wedge \bigwedge_{f \in \mathbf{Infl}} (\delta(f, \mathrm{T}) \vee_x \delta(f, \mathrm{F})))$$

is said to be the *action formula* corresponding to a. □

The subsumption result follows.

Proposition 13 Let $\Upsilon' = \langle \sigma, \mathrm{SCD}, \mathrm{OBS}\rangle$ be a scenario description in the sense of DB97, let *A* be the set of action formiae corresponding to action expressions in SCD, and define the scenario description $\Upsilon = \langle \sigma, \mathrm{OBS} \cup \mathcal{A}, \mathcal{F}\rangle$, where $\sigma = \langle \mathcal{T}, \mathcal{F}\rangle$. Then the set of models of T' is identical to the set of models of Υ.
Proof: Just compare the definitions. □

4 Complexity Results

The following result is easy.

Theorem 14 Deciding satisfiability of a scenario description is NP-hard.
Proof: NP-hardness follows, since we can express satisfiability of propositional logic formulae. □

[3]Often, the term *intended model* is used for models of scenario descriptions. However, it seems more philosophically correct to name it just *model* We shall also name the intended models in DB97 just *models.*

[4]Infinite sets of changes could express open intervals of changes, which would require some more machinery to obtain a suitable definition.

[5]Due to space limitations, we refer the reader to the original paper instead of repeating the definitions here.

NP-membership, on the other hand, is more involved to prove. For instance, the obvious method of guessing a combination of changes and verifying that this set is a member of the set of possible combinations of changes, fails, since the set of sets can be exponentially large. We need some auxiliary notions.

Definition 15 Let Γ be a set of formulae. Then define *time* (Γ) to be the set of time point expressions used in Γ. Also, for a scenario description $\Upsilon = \langle\sigma, \Gamma, \mathcal{I}\rangle$, define *time*$(\Upsilon) = time(\Gamma)$, and *changes*$(\Upsilon)$ to be the set *time* $(\Upsilon) \times \mathcal{I} \times \{\text{T}, \text{F}\}$.

Let J be an interpretation over σ and suppose that $e \in I(\Upsilon)$. For each $\langle t, f, \tau\rangle$ fi€d e, an $\alpha \in$ *time*(Υ) such that $I(\alpha) = t$ (such an α always exists); collect these tuples $\langle\alpha, f, \tau\rangle$ in the set U. Now let *synteff* be a function taking Υ, I and e, returning such an f/, that is, *synteff* $(\Upsilon, I, e) = U$ (such a function clearly exists).

The function thus makes a syntactic representation of a set of changes, relative to a scenario description and an interpretation. Such a set is said to be a *syntactic combination of changes.* □

Next, we shall show how to represent an interpretation / in terms of a set of formulae.

Definition 16 ($A(T)$, *intrep*) Let $\sigma = \langle\mathcal{T}, \mathcal{F}\rangle$ be a signature, $T \subseteq \mathcal{T}^*$, and set $A(T) = \{[\alpha]f, \alpha R\beta \mid \alpha, \beta \in T, f \in \mathcal{F}, R \in \{<, \leq, =, \geq, >\}\}$.

Let $I = \langle h, \phi\rangle$ be an interpretation over σ. Then define $intrep(I, T) = \{\gamma \in A(T) \mid I(\gamma) \neq \emptyset\}$. Note that the sizes of A and thus $intrep(I, T)$ are polynomial in the sizes of T and $\mathcal{F}$. □

Thus *intrep*(I, T) represents the interpretation /, given that only the time point expressions in T are important.

Proposition 17 Verifying for an arbitrary subset $B \subseteq A(T)$ whether B — *intrep(l,T)* for some J and T can be done in time polynomial in the size of *A(T).*

Proof: We use a result of DB97, saying that satisfiability of a set of *Horn formulae* can be solved in polynomial time. For this proof, it is enough to know that formulae in *A(T)* and negations of such formulae are Horn formulae.

Construct the set $B' = B \cup \{\neg\gamma \mid \gamma \in A(T), \gamma \notin B\}$. Now it is clear that B^* has a model / iff B — *intrep*(I, T), and the result follows. □

Corollary 18 Let / be an interpretation over $\sigma = \langle\mathcal{T}, \mathcal{F}\rangle$, $T \subseteq \mathcal{T}^*$, and let W ~ *intrep(1*, T). Then querying whether $I(\gamma) \neq \emptyset$ for $\gamma \in$ *A{T)* can be done in time polynomial in the size of *A(T).*

Proof: Just check whether $\gamma \in W$. □

Definition 19 (Annotated formula) Let $\Upsilon = \langle\sigma, \Gamma, \mathcal{I}\rangle$ be a scenario description, $\gamma = \bigwedge\Gamma$, and let the formula γ' (which is never a member of Σ) be obtained from 7 by replacing every subformula β of γ (γ is counted as a subformula of itself) by the expression $\langle\beta, E, \alpha\rangle$, for $E \subseteq \{e\}$, with some arbitrarily chosen e $\subseteq$ *changes*(Υ), and $\alpha \in$ *time*(Υ). Then γ' is said to be an *annotated formula* for Υ. Note that the size of γ' is polynomial in the size of Υ, and that γ can always easily be recovered from γ'. Then we write $\gamma ==$ *unannotate*(γ'). □

Definition 20 (Syntactic scenario interpretation, syntactic scenario model) Let $\Upsilon = \langle\sigma, \Gamma, \mathcal{I}\rangle$ be a scenario description, T = *time*(Υ), S a p $\varepsilon\langle W, \gamma'\rangle$, i t h $W \subseteq A(T)$ and $\gamma' = \langle\gamma, E, \alpha\rangle$ an annotated formula for Υ. Then S is said to be a *syntactic scenario interpretation* for Υ.

For the definition of a syntactic scenario model, we need an auxiliary definition: define the function *condeff₈(B,E)* (a syntactic variant of the function *condeff),* taking a truth value B and a set E of possible syntactic combinations of changes, returning E if B is true, and $\emptyset$ otherwise.

Furthermore, for a syntactic scenario interpretation $S = \langle W, \gamma'\rangle$, we define a predicate S on γ' as follows, letting $\oplus \in \{\wedge, \vee, \vee_x, \rightarrow, \leftrightarrow\}$, $R \in \{=, \leq, <, \geq, >\}$, $\alpha, \beta \in \mathcal{T}^*$, $\alpha, \alpha', \alpha_i, \beta \in \mathcal{T}^*$, and $\tau \in \{\text{T}, \text{F}\}$. F should be a previously unused variable.

$$
\begin{aligned}
&S(\langle\tau, E, \alpha\rangle) \Leftrightarrow \\
&\qquad E = condeff_s(\tau, \{\emptyset\}) \\
&S(\langle f, E, \alpha\rangle) \Leftrightarrow \\
&\qquad E = condeff_s([\alpha]f \in W), \{\emptyset\}) \\
&S(\langle\delta(f, \text{T}), E, \alpha\rangle) \Leftrightarrow \\
&\qquad E = \{\{\langle\alpha, f, \text{T}\rangle\}\} \\
&S(\langle\delta(f, \text{F}), E, \alpha\rangle) \Leftrightarrow \\
&\qquad E = \{\{\langle\alpha, f, \text{F}\rangle\}\} \\
&S(\langle\alpha' R\beta, E, \alpha\rangle) \Leftrightarrow \\
&\qquad E = condeff_s(\alpha' R\beta \in W), \{\emptyset\}) \\
&S(\langle\neg\langle\gamma, E_\gamma, \alpha_\gamma\rangle, E, \alpha\rangle) \Leftrightarrow \\
&\qquad E = \neg E_\gamma \wedge \alpha = \alpha_\gamma \\
&S(\langle\langle\gamma, E_\gamma, \alpha_\gamma\rangle \oplus \langle\epsilon, E_\epsilon, \alpha_\epsilon\rangle, E, \alpha\rangle) \Leftrightarrow \\
&\qquad F = E_\gamma \oplus E_\epsilon \wedge E \subseteq F \wedge \\
&\qquad (E = \emptyset \Rightarrow F = \emptyset) \wedge \alpha = \alpha_\gamma = \alpha_\epsilon \\
&S(\langle[\alpha]\langle\gamma, E_\gamma, \alpha_\gamma\rangle, E, \alpha\rangle) \Leftrightarrow \\
&\qquad E = E_\gamma \wedge \alpha = \alpha_\gamma,
\end{aligned}
$$

where the operators ~s A, V, Vₐ., —▶ and «> on sets *E\, Ei* of possible syntactic combinations of changes are defined by

$$
\begin{aligned}
\neg E_1 &= \begin{cases} \emptyset & \text{if Ei 7* 0} \\ \{\emptyset\} & \text{otherwise} \end{cases} \\
E_1 \wedge E_2 &= \{e_1 \cup e_2 \mid e_1 \in E_1, e_2 \in E_2\} \\
E_1 \vee E_2 &= \{e_1, e_2, e_1 \cup e_2 \mid e_1 \in E_1, e_2 \in E_2\} \\
E_1 \vee_x E_2 &= \{e_1 \in E_1 \mid \neg\exists e_2 \in E_2 . e_2 \subseteq_s e_1\} \cup \\
&\quad\ \{e_2 \in E_2 \mid \neg\exists e_1 \in E_1 . e_1 \subseteq_s e_2\} \\
E_1 \rightarrow E_2 &= \begin{cases} E_2 & \text{if } E_1 \neq \emptyset \\ \{\emptyset\} & \text{otherwise} \end{cases} \\
E_1 \leftrightarrow E_2 &= (E_1 \rightarrow E_2) \wedge (E_2 \rightarrow E_1),
\end{aligned}
$$

where the relation $\subseteq_s$ is defined such that $e_1 \subseteq_s e_2$ iff for each $\langle\alpha, f, \tau\rangle \in e_1$, there exists a $\langle\alpha', f, \tau\rangle \in e_2$, with $\alpha = \alpha' \in W$. The relation models inclusion of syntactic combinations of changes, taking into account which syntactically differing time points are semantically equal. Note that this relation can be computed in polynomial time, due to our assumptions.

Now, if the following conditions hold, then S is said to be a *syntactic scenario model* of Υ:

- W = *intrep(I,T)* for some interpretation I
- $S(\beta)$ is true for every subformula β of γ'
- there exists an $e \in E$ such that
  - for each $\langle \alpha, f, \tau \rangle \in e$, $[\alpha]f \in W \Leftrightarrow \tau$
  - for each $f \in \mathcal{I}$ and $\alpha, \beta \in time(\Upsilon)$ with $\alpha < \beta \in W$ such that for no α' for which $\alpha < \alpha', \alpha' < \beta \in W$, $\langle \alpha', f, \tau \rangle \in e$ holds (for some $\tau \in \{\mathrm{T}, \mathrm{F}\}$), we have that $[\beta]f \in W \Leftrightarrow [\alpha]f \in W$.

It is clear that S can be computed in polynomial time, and similarly for the remaining checks, so checking wheter a syntactic scenario interpretation is a syntactic scenario model can be checked in polynomial time. □

Theorem 21 Deciding satisfiability of a scenario description is NP-complete.

Proof (sketch): It remains to prove NP-membership, by Theorem 14. Now, we can use a syntactic scenario interpretation as a guess, and then verify whether it is a scenario model or not in polynomial time, by the previous results. The existence of a syntactical scenario model and the existence of a model of the scenario description can easily proved to be equivalent. □

5 Discussion

We have expanded the expressivity boundaries for reasoning about action in NP time. This is a proof that a polynomial-time reduction exists to propositional logic, making stochastic search procedures like GSAT applicable to the problem. The formalism presented here is clearly more expressive than e.g. *A* in all aspects except that we do not have branching time (we can handle explicit, continuous time, nondeterministic actions, "maybe" actions and so on), so the result that computational complexity of satisfiability in the two formalisms is equivalent is somewhat surprising. Then a question is: how much further can one go? Since there is no precise measure for expressivity, this is a difficult question. Moreover, some extensions of the logic presented in this paper will not prove to be NP-complete with the proof technique we have employed. A basic tool in the NP-membership proof is to find a polynomially-sized representation of an interpretation. Now, if would extend the logic to allow quantification over time points (which could represent causal rules), we could have a scenario description $\Upsilon = \langle \sigma, \Gamma, \{f\} \rangle$ where

$$\Gamma = \{\forall t.t \neq 0 \rightarrow (([t]f \rightarrow [2t]\delta(f, \mathrm{F})) \wedge ([t]\neg f \rightarrow [2t]\delta(f, \mathrm{T})))\}.$$

This scenario description has uncountably many models, so the method of representing interpretations and changes in polynomial space will fail (there are only countably many representations of whatever needs to be represented). Thus, some new proof technique would have to be employed.

6 Conclusions

We have introduced a logic of action and change which is expressive in the sense that it can represent most propositional benchmark examples in the literature, and some new examples involving parallel composition of actions, and actions that may or may not be executed. We have proved that satisfiability of a scenario in this logic is NP-complete, and that it subsumes an NP-complete logic introduced by Drakengren and Bjareland [1997]

References

[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.

[Baral and Gelfond, 1997] Chitta Baral and Michael Gelfond. Reasoning about effects of concurrent actions. *Journal of Logic Programming,* 31:85 118, 1997.

[Donerty, 1994] Patrick Donerty. Reasoning about action and change using occlusion. In *EC AI '94-*

(Drakengren and Bjareland, 1997] Thomas Drakengren and Marcus Bjareland. Reasoning about action in polynomial time. In *IJCAI '97.*

[Gelfond and Lifschitz, 1993] Michael Gelfond and Vladimir Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming,* 17:301-321, 1993.

[Giunchiglia *et* a/., 1997] Enrico Giunchiglia, G. Neelakantan Kartha, and Vladimir Lifschitz. Representing action: indeterminacy and ramifications. *Artificial Intelligence,* 95:409-438, 1997.

[Gustafsson and Doherty, 1996] J. Gustafsson and P. Doherty. Embracing occlusion in specifying the indirect effects of actions. In *KR '96.*

[Liberatore, 1997] P. Liberatore. The complexity of the language *A. Linkoping Electronic Articles in Computer and Information Science,* 2(2), July 1997. URL: http://www.ep.liu.se/ea/cis/1997/006.

[McCain and Turner, 1995] Norman McCain and Hudson Turner. A causal theorv of ramifications and qualifications. In *IJCAI '95.*

[McCarthy and Hayes, 1969] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of Artificial Intelligence. *Machine Intelligence,* 4:463 502, 1969.

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

[Selman *et* a/., 1992] Bart Selman, Hector Levesque, and David Mitchell. A new method for solving hard satisfiability problems. In *AAA! '92.*