Proceedings of the Sixteenth International Conference on
Principles of Knowledge Representation and Reasoning (KR 2018)

On the Progression of Situation Calculus Universal Theories with Constants*

Marcelo Arenas,[1,3] **Jorge A. Baier,**[1,3] **Juan S. Navarro,**[1,3] **Sebastian Sardina**[2]
[1] Pontificia Universidad Católica de Chile
[2] RMIT University, Australia
[3] Instituto Milenio Fundamentos de los Datos, Chile
{marenas, jabaier}@ing.puc.cl, jsnavar1@uc.cl, sebastian.sardina@rmit.edu.au

Abstract

The progression of action theories is an important problem in knowledge representation. Progression is second-order definable and known to be first-order definable and effectively computable for restricted classes of theories. Motivated by the fact that universal theories with constants (UTCs) are expressive and natural theories whose satisfiability is decidable, in this paper we provide a thorough study of the progression of situation calculus UTCs. First, we prove that progression of a (possibly infinite) UTC is always first-order definable and results in a UTC. Though first-order definable, we show that the progression of a UTC may be infeasible, that is, it may result in an infinite UTC that is not equivalent to any finite set of first-order sentences. We then show that deciding whether there is a feasible progression of a UTC is undecidable. Moreover, we show that deciding whether a sentence (in an expressive fragment of first-order logic) is in the progression of a UTC is coNEXPTIME-complete, and that there exists a family of UTCs for which the size of every feasible progression grows exponentially. Finally, we discuss resolution-based approaches to compute the progression of a UTC. This comprehensive analysis contributes to a better understanding of progression in action theories, both in terms of feasibility and difficulty.

Introduction

Action theories are knowledge representation formalisms for modeling and reasoning about dynamic systems, an important aspect of commonsense reasoning. To represent them, a plethora of action languages have been proposed; for example, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ (Gelfond and Lifschitz 1998), with their English like syntax, the narrative-based Event Calculus (Kowalski and Sergot 1986; Shanahan 1999), state-centric Fluent Calculus (Thielscher 1999), the extensible Features and Fluents (Sandewall 1994), and the Situation Calculus (Pirri and Reiter 1999; Reiter 2001).

At the very least, action languages provide means to specify actions' effects and non-effects, thus embracing some type of solution to the frame problem (McCarthy and Hayes 1969; Shanahan 1997). They also provide a means to specify the "initial knowledge base" describing the state of the world before anything has occurred.

A fundamental problem in reasoning about action and change is the *progression problem* (Lin and Reiter 1994), that is, the problem of evolving the representation of the current state as actions occur. The problem has lately become more relevant, as we see KR technologies as modules of situated intelligent agents and robots that not only reason, but also act and observe the environment (Ghallab, Nau, and Traverso 2014). Its solution is also key to enable other agent technologies, such as planning and high-level program execution (Reiter 1993; De Giacomo et al. 2009; Ghallab, Nau, and Traverso 2014). In fact, it is generally recognized that progression is mandatory to efficiently address the important *projection task* (Vassos and Levesque 2008), namely, the task of determining whether some condition holds after a given set of actions have been performed.

This paper contributes to the understanding of the progression problem in action theories by analyzing its feasibility and computational difficulty for a fundamental special case of Basic Action Theories (BATs) (Reiter 1991; Pirri and Reiter 1999). BATs are Situation Calculus axiomatizations of a certain shape that can be built following a prescribed methodology for extracting the necessary axioms. Importantly, the methodology yields a parsimonious and effective solution to the frame problem within classical logic. BATs have been studied at depth and been shown to enjoy a number of good properties, notably, the ability to reduce the projection task to first-order theorem proving against the initial state representation. They have also been shown elaboration tolerant via multiple extensions, such as time (Pinto 1994), high-level programs (Levesque et al. 1997; De Giacomo, Lespérance, and Levesque 2000; Sardina et al. 2004), ramifications (Pinto 1999; McIlraith 2000), concurrency and natural actions (Reiter 1996), and even probabilistic reasoning (Vassos and Levesque 2008).

Unfortunately, though, the progression of a BATs is *not*, in general, first-order definable (Lin and Reiter 1994; 1997; Vassos and Levesque 2008)—an update after an action may not yield a (successor) BAT. Thus, understanding first-order definability and effectiveness of restricted classes of BATs becomes paramount if we hope situated intelligent agents

*We would like to thank the anonymous referees for their helpful comments. Arenas, Baier, and Navarro were partially funded by the Instituto Milenio Fundamentos de los Datos. Navarro was also partially funded by the CONICYT-PFCHA/Doctorado Nacional/2018-21182042 scholarship.
Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

to be equipped with such type of theories, see, e.g., (Lin and Reiter 1995; Vassos, Lakemeyer, and Levesque 2008; Liu and Lakemeyer 2009; Vassos and Sardina 2011).

Motivated by the fact that so-called *universal theories with constants* (UTCs)—part of the *Bernays-Schönfinkel class* (Bernays and Schönfinkel 1928)—are expressive and natural theories whose satisfiability is decidable, we set to study the progression of BATs within such class, which we call UTC-BATs. It turns out that UTC-BATs can express complex causal laws like *"opening the gripper will cause every object being held to fall and break."* Importantly, UTC-BATs remains in the realms of classical logic. Thus, our work distinguishes from that of Liu and Lakemeyer (2009) on progression of proper$^+$ theories that rely on standard names to characterize the universe of discourse.

The results obtained are mixed, but overall demonstrate how difficult the progression of sufficiently rich theories is. Concretely, we first prove that progression of a (possibly infinite) UTC-BAT is always first-order definable and yields a UTC-BAT. This is a positive result that eludes the main negative result of progression in BATs (Vassos and Levesque 2008). However, such progression may be *infeasible*, in that it may turn out to be intrinsically infinite (i.e., not equivalent to any finite version defined in terms of first-order logic).[1] Given this, we demonstrate that deciding whether there is a feasible progression is undecidable. When it comes to computational complexity, we show that deciding whether a sentence (in an expressive fragment of first-order logic) is in the progression of a UTC-BAT is CONEXPTIME-complete, and that there exists a family of UTC-BATs for which the sizes of the progressions grow exponentially. We close by discussing resolution-based approaches to actually compute the progression of a UTC-BAT. All in all, our results provide a comprehensive map of feasibility and complexity for evolving a natural and expressive case of BATs.

Background

This section presents the background for the rest of the paper. We assume familiarity with first-order (FO) logic.

Universal Formulae with Constants

An FO formula is a $\exists^*\forall^*$-FO formula if it is of the form:

$$\exists x_1 \cdots \exists x_k \forall y_1 \cdots \forall y_m \, \phi, \tag{1}$$

where ϕ is a quantifier-free formula. The class of $\exists^*\forall^*$-FO sentences not mentioning function symbols except for constants (which are 0-ary functions) is known as the *Bernays-Schönfinkel class* (Bernays and Schönfinkel 1928). The problem of determining whether a sentence in the Bernays-Schönfinkel class has a model is NEXPTIME-complete (Bernays and Schönfinkel 1928; Lewis 1980). As this class is closed under conjunction, the complexity is the same for a finite set of sentences in the Bernays-Schönfinkel class.

Membership in NEXPTIME follows from two facts. The first fact is the *finite-domain property*: if a formula of the form (1) in the Bernays-Schönfinkel class is satisfiable, then it has a model whose domain size is at most $k + \ell$, where ℓ is the number of constants occurring in ϕ. Indeed, let $\Psi = \exists x_1 \cdots \exists x_k \forall y_1 \cdots \forall y_m \phi$ and assume Ψ has a model $\mathbb{M}$ whose domain size is greater than $k + \ell$. To construct a model $\mathbb{M}'$ for Ψ of domain size at most $k + \ell$, we assume that μ is an assignment for the variables $\{x_1, \ldots, x_k\}$ such that $\mathbb{M}, \mu \models \forall y_1 \cdots \forall y_m \, \phi$ (by assuming $\mathbb{M} \models \Psi$, we know μ exists). Now assuming that $c_1, \ldots, c_\ell$ is the set of constants occurring in ϕ, we define $\mathbb{M}'$ with domain $D = \{\mu(x_1), \ldots, \mu(x_k), c_1^{\mathbb{M}}, \ldots, c_\ell^{\mathbb{M}}\}$, such that for each predicate P of arity n and every $\vec{o} \in D^n$, it holds that $\vec{o} \in P^{\mathbb{M}'}$ if and only if $\vec{o} \in P^{\mathbb{M}}$. Thus, $\mathbb{M}'$ is the substructure of $\mathbb{M}$ induced by D. Then, we have that $\mathbb{M}', \mu \models \forall y_1 \cdots \forall y_m \, \phi$, as this is a universal formula and $\mathbb{M}'$ is an induced substructure of $\mathbb{M}$ (Hodges 1997), from which we conclude that $\mathbb{M}' \models \Psi$. The second fact is that a model with domain size at most $k + \ell$ can be of exponential size, as the arities of the predicates mentioned in ϕ are not fixed. Hence, by constructing a non-deterministic algorithm that simply guesses such a model we can conclude that the problem is in NEXPTIME.

A special case of $\exists^*\forall^*$-FO formulae is *universal formulae with constants*, formulas of the form $\forall y_1 \cdots \forall y_m \, \phi$, where ϕ does not have any quantifiers or function symbols except for constants. A universal theory with constants (UTC) is a set of universal formulae with constants. If $\Sigma \cup \{\varphi\}$ is a UTC, then the problem of verifying whether $\Sigma \models \varphi$ is CONEXPTIME. In particular, the membership in CONEXPTIME holds since the problem of deciding whether $\Sigma \models \varphi$ can be reduced to verifying whether $\Sigma \cup \{\neg\varphi\}$ is unsatisfiable, which in turn can be decided in CONEXPTIME as $\Sigma \cup \{\neg\varphi\}$ is a set of Bernays-Schönfinkel sentences.

The Situation Calculus

The Situation Calculus (Reiter 2001) is a logical language designed for representing dynamically changing worlds whose changes are the result of *actions*. Denoted by $\mathcal{L}_{sitcalc}$ it is a many-sorted second-order language with equality. Sort *situation* are finite sequences of actions: the empty sequence is denoted by the constant S_0, and the situation that results from performing action a in situation s is denoted $do(a, s)$. Sort *object* is a catch-all sort for all objects in the world.

In $\mathcal{L}_{sitcalc}$, ***fluents*** describe the dynamic aspects of the world. A (relational) fluent is a predicate whose last argument is a situation, and whose truth value can change from situation to situation. For example, fluent *Broken*(x, s) can be used to denote that object x is broken in situation s.

Since actions need not be always executable, predicate *Poss*(a, s) is used to state that action a is executable in situation s. A formula φ of $\mathcal{L}_{sitcalc}$ is ***uniform*** in situation term σ (Reiter 2001) if the only situation term mentioned in φ is σ, neither *Poss*, $\sqsubset$, nor equalities between situations are mentioned, and there is no quantification over situations.

Reiter's *basic action theories* (BATs) are well-studied theories in $\mathcal{L}_{sitcalc}$ of the form $\mathcal{D} = \mathcal{D}_{S_0} \cup \mathcal{D}_{poss} \cup \mathcal{D}_{ssa} \cup \mathcal{D}_{una} \cup \Sigma$, where:

1. $\mathcal{D}_{S_0}$ is a set of formulae uniform in S_0, stating what holds true in the initial situation (the initial knowledge base);
2. $\mathcal{D}_{poss}$ is the set of *precondition axioms*, one per action A,

[1] Notice that Vassos and Levesque (2008) showed that even an infinite set of first-order sentences will not suffice for some BATs.

of the form $Poss(A(\vec{x}), s) \equiv \Pi_A(\vec{x}, s)$, where $\Pi_A(\vec{x}, s)$ is a formula that is uniform in s, stating when an action is legally executable in a situation;

3. $\mathcal{D}_{ssa}$ is the set of *successor state axioms*, one per fluent F, of the form[2] $F(\vec{x}, do(a, s)) \equiv \Phi_F(\vec{x}, a, s)$, where Φ_F is a formula uniform in s, characterizing how F evolves in situation s when action a is performed;
4. $\mathcal{D}_{una}$ is the set of *unique names axioms* for actions, stating that actions with different names are different; in addition, for every action a we write $a(\vec{x}) = a(\vec{y}) \supset \vec{x} = \vec{y}$.
5. Σ is a set of domain-independent foundational axioms characterizing the set of possible situations as sequences of actions from S_0.

Notice that non-quantified variables in the previous axioms are assumed to be universally quantified at the outermost level. Throughout the paper, we make the same assumption for each formula with free variables.

Progression of BATs

Progressing a BAT involves *updating its the initial knowledgebase* after some actions have been performed. More precisely, given a BAT $\mathcal{D}$ and a ground action term α that is legally executable in S_0 (i.e., $Poss(\alpha, S_0)$ holds), the task is to replace $\mathcal{D}_{S_0}$ in $\mathcal{D}$ with a set of sentences $\mathcal{D}_{do(\alpha,S_0)}$ (the new initial KB after α's execution), so that the original theory $\mathcal{D}$ and the updated theory $(\mathcal{D} \setminus \mathcal{D}_{S_0}) \cup \mathcal{D}_{do(\alpha,S_0)}$ are equivalent with respect to how they describe the situation $do(\alpha, S_0)$ as well as all future situations from $do(\alpha, S_0)$ (Lin and Reiter 1997). For legibility, we will use S_α and $\mathcal{D}_\alpha$ to denote $do(\alpha, S_0)$ and $\mathcal{D}_{do(\alpha,S_0)}$, respectively.

The notion of progression is defined in terms of legally executable actions. For the sake of simplicity, and without loss of generality, we assume from now on that all actions are possible always, that is, we assume that $Poss(A(\vec{x}), s) \equiv \mathtt{True} \in \mathcal{D}_{poss}$, for each action A. We also follow the definition of the so-called *strong progression* of (Vassos, Lakemeyer, and Levesque 2008; Vassos and Levesque 2013). When $F_1, \ldots, F_n$ are relational fluents, $Q_1, \ldots, Q_n$ are second-order (non-fluent) predicate variables, and ϕ is an FO-formula, we write $\phi\langle\vec{F} : \vec{Q}\rangle$ to denote the formula that results from replacing any fluent atom $F_i(t_1, \ldots, t_n, \sigma)$ in ϕ, where σ is a situation term, with atom $Q_i(t_1, \ldots, t_n)$.

Definition 1 (Vassos and Levesque 2013)**.** *Let $\mathcal{D}$ be a BAT over fluents $F_1, \ldots, F_n$ and with $\mathcal{D}_{S_0}$ being a finite set of sentences whose conjunction is denoted by ϕ_0. Let α be a ground action term and $Q_1, \ldots, Q_n$ be predicate variables. Then, $Pro(\mathcal{D}, \alpha)$ is the following second-order sentence uniform in S_α:*

$$\exists\vec{Q}\,(\phi_0\langle\vec{F} : \vec{Q}\rangle \wedge \bigwedge_{i=1}^{n} \forall\vec{x}_i\, F_i(\vec{x}_i, S_\alpha) \equiv \Phi_{F_i}(\vec{x}_i, \alpha, S_0)\langle\vec{F} : \vec{Q}\rangle).$$

We say that a set of formulae $\mathcal{D}_\alpha$ uniform in situation $do(\alpha, S_0)$ is a strong progression *of $\mathcal{D}$ w.r.t. action α* iff *$\mathcal{D}_\alpha \cup \mathcal{D}_{una}$ is logically equivalent*[3] *to $\{Pro(\mathcal{D}, \alpha)\} \cup \mathcal{D}_{una}$.*

In their seminal work, Lin and Reiter (1997) showed that there are cases in which there does not exist an FO progression and a second-order one is required. Finding interesting cases for which FO progressions $\mathcal{D}_\alpha$ can be found is an ongoing challenge. In fact, a natural alternative of the notion of progression amounts to taking a strong enough set that allows the entailment of all FO-sentences uniform in S_α that are entailed by the original theory:

Definition 2. *Let $\mathcal{D}$ be a BAT and α a ground action term. A set $\mathcal{F}_\alpha$ of FO-sentences uniform in S_α is a* weak progression *of $\mathcal{D}$ w.r.t. α iff for all FO-sentence ϕ uniform in S_α, it holds that $\mathcal{F}_\alpha \cup \mathcal{D}_{una} \models \phi$ iff $\mathcal{D} \models \phi$.*

Vassos and Levesque (2008) proved that the weak progression is not always correct: there is a BAT $\mathcal{D}$ and an action α such that there is an FO-formula entailed by $\mathcal{D}$ but not by $(\mathcal{D} \setminus \mathcal{D}_{S_0}) \cup \mathcal{F}_\alpha$. Note that, by definition of weak progression, such a formula cannot be uniform in S_α, and has to refer to future situations from S_α. Nonetheless, weak progression happens to be enough for a wide set of cases, including STRIPS and simple projection queries (Lin and Reiter 1997), and interesting cases of generalized projection on queries about the future (Vassos and Levesque 2013).

For this reason, several restrictions on the theories have been proposed so that the updated KB is indeed FO definable. Vassos, Lakemeyer, and Levesque (2008) showed that progression for *local-effect* actions is FO definable. Local-effect actions are those that can only affect objects directly specified in its arguments. Then, Liu and Lakemeyer (2009) extended that result to *normal actions*, which are able to affect objects not mentioned in their arguments as long as these are specified by information that exists in the KB in a particular form. In addition, they showed that progression of local-effect and normal actions can actually be efficiently computed for action theories in the proper$^+$ KB fragment (Lakemeyer and Levesque 2002). In turn, Vassos and Sardina (2011) studied progression of *range-restricted* theories, which are able to handle actions that may not be local-effect or normal but whose (range of) effects can be "bounded" at execution time (e.g., a robot "moving forward"). Using classical database evaluation techniques, they described how to obtain FO and finite progressions that are correct for the fundamental class of conjunctive queries.

Feasible Progression

As we saw before, weak progression is a restricted form of progression that is sometimes correct, that is, equivalent to strong progression. A clear advantage of a weak progression is that it is a set of FO-sentences and, thus, admits various forms of automated reasoning. Additionally, an FO progression can be used to create a new BAT from the original one, something not possible when no FO progression exists.

[2]Following Reiter (2001), we use notation $\vec{x}$ to mean different things depending on the context. $F(\vec{x}, s)$ is used as a shorthand for $F(x_1, \ldots, x_n, s)$, where $n + 1$ is F's arity. $\mathcal{Q}\vec{x}\,\varphi$ is a shorthand for $\mathcal{Q}x_1 \cdots \mathcal{Q}x_n\,\varphi$, with $\mathcal{Q} \in \{\forall, \exists\}$.

[3]Whenever we say that two formulae are logically equivalent we assume that the logical symbol $=$ is always interpreted as the true identity.

Nevertheless, as we will see below, there are cases in which the weak progression, even though provably equivalent to a strong progression, may not be computable because it might require an *infinite* set of FO-sentences $\mathcal{F}_\alpha$. This can happen even if $\mathcal{D}_{S_0}$ is finite (see Corollary 1). This motivates us to introduce another notion of progression, which we call *feasible*, that enforces finiteness.

Definition 3. *Let $\mathcal{D}$ be a BAT and α a ground action term. Set $\mathcal{F}_\alpha$ is a* feasible progression *of $\mathcal{D}$ w.r.t. α iff (1) $\mathcal{F}_\alpha$ is a strong progression of $\mathcal{D}$ w.r.t. α; and (2) $\mathcal{F}_\alpha$ is a finite set of FO-sentences.*

When a feasible progression for $\mathcal{D}$ w.r.t. α exists, we say that the progression of $\mathcal{D}$ w.r.t. α is feasible; otherwise, we say its progression is infeasible. The following result establishes a formal relation among the three notions progression.

Proposition 1. *Let $\mathcal{D}$ be a BAT and α a ground action term. If the strong progression of $\mathcal{D}$ w.r.t. α is equivalent to a finite set Δ of* FO*-sentences, then Δ is a strong, weak, and feasible progression of $\mathcal{D}$ w.r.t. α.*

Proof. Let Δ be a finite set of FO-sentences equivalent to a strong progression $\mathcal{D}_\alpha$ of $\mathcal{D}$ w.r.t. α. This set satisfies that $\Delta \cup \mathcal{D}_{una} \equiv \mathcal{D}_\alpha \cup \mathcal{D}_{una}$. By definition of strong progression, $\mathcal{D}_\alpha \cup \mathcal{D}_{una} \equiv \{Pro(\mathcal{D}, \alpha)\} \cup \mathcal{D}_{una}$, so by transitivity of logical equivalence, $\Delta \cup \mathcal{D}_{una} \equiv \{Pro(\mathcal{D}, \alpha)\} \cup \mathcal{D}_{una}$, which proves that Δ is a strong progression. By hypothesis Δ is a finite set of FO-sentences, so it is a feasible progression by definition. Finally, given an FO-sentence ψ uniform in S_α, it holds that $\Delta \cup \mathcal{D}_{una} \models \psi$ because Δ is a strong progression (Vassos and Levesque 2013; Lin and Reiter 1997). Hence, we have that Δ is also a weak progression. $\square$

Universal BATs with Constants

Motivated by the fact that universal theories with constants are expressive and natural theories whose satisfiability is decidable, we analyze the properties regarding progression that can be obtained when restricting BATs to have this form.

We start off by defining Universal Basic Action Theories with Constants (UTC-BATs) which are BATs in which the axioms relevant to progression are in the required form.

Definition 4. *A BAT $\mathcal{D}$ is a* Universal BAT with Constants *(UTC-BAT) if for any ground action term α, $\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{ssa}[\alpha, S_0]$ can be simplified to UTCs after replacing $do(\alpha, S_0)$ with S_α.*

In the definition above, $\mathcal{D}_{ssa}[\alpha, S_0]$ is the result of substituting a by α and s by S_0 in every successor state axiom $F(\vec{x}, do(a, s)) \equiv \Phi_F(\vec{x}, a, s)$ in $\mathcal{D}_{ssa}$ (Lin and Reiter 1997). Moreover, the simplification process involves using the unique name axioms with the objective of eliminating action terms that may occur in the formula. To clarify how this simplification works, let us consider a theory of action with the following two successor state axioms:

$$Broken(x, do(a, s)) \equiv \\ (a = dropAll \wedge Holding(x, s)) \vee Broken(x, s); \quad (2)$$

$$Holding(x, do(a, s)) \equiv \\ a = pick(x) \vee (Holding(x, s) \wedge a \neq dropAll) \quad (3)$$

Note that $\mathcal{D}_{ssa}[\alpha, S_0]$, for every action term α, can be simplified in such a way that almost all action terms are eliminated, except for $do(\alpha, S_0)$. In particular, if α is $pick(B)$, then $\mathcal{D}_{ssa}[\alpha, S_0]$ is simplified into:

$$Broken(x, do(pick(B), S_0)) \equiv Broken(x, S_0)$$
$$Holding(x, do(pick(B), S_0)) \equiv (B = x \vee Holding(x, S_0))$$

A relation between UTC-BATs and other classes of BATs studied in the literature is sometimes possible to establish. Context-free BATs (Lin and Reiter 1997), provided that the situation independent sentences in $\mathcal{D}_{S_0}$ are universal formulae with constants, are also UTC-BATs. As a consequence, classical STRIPS (Fikes and Nilsson 1971) as well as its open-world version (Lin and Reiter 1997) are cases of UTC-BATs, since their successor state axioms and initial theories are further restrictions to those in context-free theories.

On the other hand, UTC-BATs are just different from local-effect (Vassos, Lakemeyer, and Levesque 2008), normal-action (Liu and Lakemeyer 2009), and even range-restricted (Vassos and Sardina 2011) based theories. For example, the action of moving from location *src* to destination *dest* whose effect is being at location *dest*, provided there is enough fuel to cover the distance, is a local-effect action, and thus normal and range-restricted, but it cannot be represented with a universal successor state axioms for fluent $AtLoc(x, s)$, as it includes a context condition of the form $\exists l.FuelLevel(l, s) \wedge MinFuel(src, dest, l)$ in the right hand side. In turn, the successor state axioms (2)–(3) would not yield a local-effect theory, as affected object x is not mentioned in the action, or even a normal theory, since dropping all objects affect both *Broken* as well as the contextual *Holding* predicate. The above axioms would also not be range-restricted if there is incomplete information on what is being held. Nonetheless, the above axioms, even when the current state has incomplete information about predicate $Holding(x, s)$ would indeed be UTC axioms. Thus, in general, these notions are incomparable against UTC-BATs.

The comparison with proper$^+$ BATs, for which Liu and Lakemeyer (2009) have provided good progression properties, is more subtle. Syntactically, proper$^+$ and UTC theories appear almost identical, as any UTC $\forall\phi$ formula could be rewritten as a set of $\forall$-clauses formulae by transformation to CNF, albeit with an exponential growth. Also, successor state axioms in UTC-BATs are indeed "essentially quantifier-free" (i.e., no quantification on the right-hand side), thus they can be defined as proper$^+$ KBs (Liu and Lakemeyer 2009, Proposition 5.11). However, proper$^+$ KB assumes a language with *standard names* (Levesque and Lakemeyer 2001), a set of infinitely many different constants representing all the individuals in the universe of discourse. This significantly simplifies the semantics (and handling of quantification), but it either requires second-order axioms or non-classical semantics, under which, for example, compactness does not apply. In contrast, and in the spirit of Reiter (2001)'s work, our restriction of BAT to UTC formulae aims to remain in the classical setting (and understand its consequences!). The issue of remaining classical is also relevant in that we will sometimes be interested in infinite (progressed) theories. Because of all this, we are not able to directly import Liu and Lakemeyer (2009)'s positive results (Theorems 5.12 and 5.13) to UTC-BATs.

Proviso. In what follows, we study the progression of UTC-BATs. It is important to notice that so far we have not imposed any restrictions on the cardinality of BATs and progressions, except for the finiteness constraint imposed in the case of feasible progressions. In the rest of the paper, we consider finite UTCs and, in particular, finite UTC-BATs, unless we explicitly say otherwise.

Defining the Progressing of a UTC-BAT

The first theoretical result that we present establishes that the strong progression of a UTC-BAT is a (possibly infinite) UTC, and, thus, always definable in FO.

Theorem 1. *Given a UTC-BAT with constants $\mathcal{D}$ and a ground action α, every strong progression of $\mathcal{D}$ w.r.t. α is equivalent to a possibly infinite UTC. Moreover, the same result holds if $\mathcal{D}_{S_0}$ is an infinite UTC.*

Proof. (Sketch) Assume that $\mathcal{L}_0$ is the language of $\mathcal{D}$ and that $F_1, \dots, F_n$ are the fluents occurring in $\mathcal{D}$. Let $\mathcal{L}_1$ be $\mathcal{L}_0$ augmented with fresh predicate symbols $Q_1, \dots, Q_n$ such that the arity of each Q_i is equal to the arity of F_i minus one. Then define a set of FO-formulae Δ by mimicking the definition of strong progression (see Definition 1), but without including second-order quantification over predicates $Q_1, \dots, Q_n$:

$$\Delta = \{\phi\langle\vec{F} : \vec{Q}\rangle \mid \varphi \in \mathcal{D}_{S_0}\} \cup \mathcal{D}_{una} \cup \bigcup_{i=1}^{n} \{\forall \vec{x}_i\, F_i(\vec{x}_i, S_\alpha) \equiv \Phi_{F_i}(\vec{x}_i, \alpha, S_0)\langle\vec{F} : \vec{Q}\rangle\}.$$

Given a $\mathcal{L}_0$-structure $\mathbb{M}_0$, we say that a $\mathcal{L}_1$-structure $\mathbb{M}_1$ is an $\mathcal{L}_1$-extension of $\mathbb{M}_0$, if $\mathbb{M}_0$ is an induced substructure of the restriction of $\mathbb{M}_1$ to $\mathcal{L}_0$.

By definition, $\mathcal{F}_\alpha$ is a (strong) progression of $\mathcal{D}$ w.r.t. α if for every $\mathcal{L}_0$-structure $\mathbb{M}_0$, $\mathbb{M}_0 \models \mathcal{F}_\alpha$ iff there is a model $\mathbb{M}$ of $\mathcal{D}$ such that $\mathbb{M}_0 \sim_{S_\alpha} \mathbb{M}$. Also, (Kreisel and Krivine 1976, Chapter 3, Theorem 12) implies that for every $\mathcal{L}_0$-structure $\mathbb{M}_0$, $\mathbb{M}_0 \models \mathcal{F}_\alpha$ iff it has an $\mathcal{L}_1$-extension $\mathbb{M}_1$ such that $\mathbb{M}_1 \models \Delta$. Then, it is sufficient to prove that the right parts of the equivalences are equivalent.

Let $\mathbb{M}_0$ be an arbitrary $\mathcal{L}_0$-structure, and $\mathbb{M}$ a model of $\mathcal{D}$ such that $\mathbb{M}_0 \sim_{S_0} \mathbb{M}$. Let $\mathbb{M}'_0$ and $\mathbb{M}'$ be the $\mathcal{L}_1$-extensions of $\mathbb{M}_0$ and $\mathbb{M}$ which satisfy $Q(\vec{t})$ iff the original structures satisfy $F(\vec{t}, S_\alpha)$. As $\mathbb{M} \models \mathcal{D}$, by construction $\mathbb{M}' \models \Delta$, but $\mathbb{M} \sim_{S_\alpha} \mathbb{M}_0$, so $\mathbb{M}'$ and $\mathbb{M}'_0$ agree on every Q_i, which implies that $\mathbb{M}'_0 \models \Delta$. To prove the opposite direction, assume that $\mathbb{M}'_0$ is a $\mathcal{L}_1$-extension of $\mathbb{M}_0$ which satisfies Δ. Then, this model can be transformed, using that $\mathcal{D}$ is UTC, to a model $\mathbb{M}$ of $\mathcal{D}$ such that $\mathbb{M}_0 \sim_{S_\alpha} \mathbb{M}$.

Notice that $\mathcal{F}_\alpha$ may be an infinite UTC. Besides, this theorem is proved without assuming that $\mathcal{D}_{S_0}$ is finite, so the same result holds if $\mathcal{D}_{S_0}$ is a infinite UTC. □

As stated, UTC-BATs capture theories whose successor state axioms are not local-effect, normal, or range-restricted, like those shown in (2)–(3). Thus, an important consequence of Theorem 1 is that progression is FO *definable* for a new syntactic class of expressive and natural BATs.

FO-definability is, of course, desirable, but a relevant practical question is whether the progression of a UTC-BAT is feasible. Below, in Corallary 1, we show that this, unfortunately, is not the case, by giving an example of a UTC-BAT whose FO progression is an *infinite* UTC. But before that, we show that the problem of determining whether the progression of a UTC-BAT is feasible is undecidable.

Theorem 2. *The problem of verifying, given a UTC-BAT $\mathcal{D}$ and a ground action α, whether there exists a feasible progression of $\mathcal{D}$ w.r.t. α is undecidable.*

Proof. (Sketch) $\mathcal{D}_{S_0}$ encodes a theory that describes the execution of a Turing machine M on an empty string. In the encoding, objects are used to represent both time steps and positions of the tape. To achieve this, we define a fluent $F(x, y, S_0)$ that expresses that y is the successor of x. It is not hard to define F in terms of a linear order (see the proof of Corollary 1 for a reference). Furthermore, the encoding uses fluent $E_q(x, S_0)$ to express that at time step x, the state of the machine is q, and fluent $T_c(x, y, S_0)$ to express that symbol c is in the tape position given by y in time step x. An analogous predicate H is defined for the position of the head in a time step. The encoding of the machine is a conjunction of several formulae, representing different aspects of the machine. These include the initial configuration, a formula encoding that the machine can be in at most one state in every time instance:

$$\bigwedge_{q \in Q} \forall x \left(E_q(x, S_0) \supset \bigwedge_{q' \in Q \,:\, q' \neq q} \neg E_{q'}(x, S_0) \right),$$

and similar formulae for cell contents and head positions. It also includes the transition function, for example, the transition given by $\delta(q, b) = (q', c, \leftarrow)$ is encoded as:

$$\begin{gathered} \forall x \forall y \forall w \forall z\, ((E_q(x, S_0) \wedge H(x, y, S_0) \wedge T_b(x, y, S_0) \wedge \\ F(x, w, S_0) \wedge F(z, y, S_0)) \supset \\ (E_{q'}(w, S_0) \wedge H(w, z, S_0) \wedge T_c(w, y, S_0))), \end{gathered}$$

and a formula indicating that in every time instance x, if the head is not in a position y and M is in a state q that is not the final state, then in the following time instance the content of the cell in position y must be the same:

$$\begin{gathered} \forall x \forall y \forall z \bigwedge_{b \in \{0,1,\mathrm{B},\vdash\}} \Bigg(\Bigg(\bigvee_{q \in Q \,:\, q \neq q_f} F(y, z, S_0) \wedge E_q(x, S_0) \\ \wedge \neg H(x, y, S_0) \wedge T_b(x, y, S_0) \Bigg) \supset T_b(x, z, S_0) \Bigg). \end{gathered}$$

There is a successor state axiom $E_q(x, do(a, s)) \equiv E_q(x, s) \wedge a \neq A$, for every state q different from the final state q_f, and $E_{q_f}(x, do(a, s)) \equiv E_{q_f}(x, s)$. In addition successor state axioms are such that fluents L and F in $do(A, S_0)$ are defined exactly as in S_0. As a consequence of this definition, after performing action A the theory intuitively "remembers" the time step at which the machine accepted, it remembers the extensions of F and

L, but "forgets" all the other information contained in S_0. Now we establish that M accepts the empty string (an undecidable property) if and only if the progression with respect to $do(A, S_0)$ is definable with a single first-order logic formula. Indeed, if the machine accepts the empty string in n steps, then the progression can be written with a single first-order logic formula, simply encoding that fact that $E_{q_f}(x, do(A, S))$ holds if x is the n-th time step, but does not hold if x is a time step occurring before the n-th time step. If the machine does not accept the empty string, then the progression must establish that in every time step x that is *finitely reachable* from the initial time step it holds that $\neg E_q(x, do(A, S_0))$. But finite reachability is not expressible by a single first-order logic formula (Hodges 1997). $\square$

As a corollary, we obtain that a UTC-BAT may not have a feasible progression.

Corollary 1. *There is a UTC-BAT $\mathcal{D}$ and a ground action α such that no feasible progression of $\mathcal{D}$ w.r.t. α exists.*

Proof. Though this is a consequence of Theorem 2, here we provide a simpler counterexample. First, $\mathcal{D}_{S_0}$ defines a total order between the objects of the domain using a predicate L. As such $L(x, y, s)$ can be understood as saying x is ordered before y in s. In addition, it defines a fluent F, in terms of L, as a successor for elements of L.

$$\forall x \neg L(x, x, S_0),$$
$$\forall x \forall y \forall z ((L(x, y, S_0) \wedge L(y, z, S_0)) \to L(x, z, S_0)),$$
$$\forall x \forall y \, (L(x, y, S_0) \vee L(y, x, S_0) \vee x = y),$$
$$\forall x \forall y \, (F(x, y, S_0) \to L(x, y, S_0)),$$
$$\forall x \forall y \forall z \, (F(x, y, S_0) \to \neg (L(x, z, S_0) \wedge L(z, y, S_0))).$$

Notice that F does not need to mention all the elements occurring in L. Moreover, $\mathcal{D}_{ssa}$ is such that action A empties predicate L maintaining the contents of F:

$$L(x, y, do(a, s)) \equiv L(x, y, s) \wedge a \neq A,$$
$$F(x, y, do(a, s)) \equiv F(x, y, s).$$

Observe that since L is a total order, no cycles can exist through pairs of objects in L in S_0. As such, F is the disjoint union of some (acyclic) successor relations in S_0 (or, in other words, F in S_0 is a graph formed by the disjoint union of some simple paths). Now assume that $\mathcal{D}_A$ is an FO progression or our theory w.r.t. A. Observe that because L is empty in $do(A, S_0)$, the axioms that define F in $\mathcal{D}$ would not benefit from using L, and thus may only refer to predicate F. It is well-known, however, that no FO-sentence can express the property that a binary predicate is the disjoint union of some (acyclic) successor relations (Hodges 1997; Libkin 2004). $\square$

Consider now progressing UTC-BATs for which feasible progressions do exist. A natural question that arises is: *can such a feasible progression be represented by a UTC?* Note that Theorem 1 does not guarantee that the answer is "yes," since it only establishes that there exists a strong progression that is represented by a possibly infinite UTC, but it does not guarantee that such a UTC is finite when a feasible progression exists. The following positive result establishes that whenever a feasible progression exists, then it can be represented by a UTC.

Theorem 3. *Given a UTC-BAT $\mathcal{D}$ and a ground action term α, if a feasible progression of $\mathcal{D}$ w.r.t. α exists, then such a progression can be represented as a UTC.*

Note that, according to our proviso, UTCs are finite by default. As such, the UTC guaranteed in Theorem 3 is *finite*, as opposed to the UTC obtained in Theorem 1.

Before proving theorem 3, we need the following lemma. This lemma is proved using properties of the sequent calculus for first-order logic, that can be found in standard books on proof theory, such as (Takeuti 1987).

Lemma 1. *Let $\mathcal{D}$ be a UTC-BAT, and α a ground action term. If φ is uniform in $do(\alpha, S_0)$ and $\mathcal{D} \models \varphi$, then there exists an universal* FO*-sentence with constants ψ such that ψ is uniform in $do(\alpha, S_0)$, $\mathcal{D} \models \psi$ and $\psi \models \varphi$.*

Proof. (Sketch) As $\mathcal{D} \models \varphi$, and φ is uniform in S_α, we have that $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa}[\alpha, S_0] \cup \mathcal{D}_{una} \models \varphi$. Then, the sequent $\Gamma_e, \mathcal{D}_{S_0}, \mathcal{D}_{ssa}[\alpha, S_0], \mathcal{D}_{una} \Rightarrow \varphi$, where Γ_e contains the equality axioms is true, hence provable in **LK**. This sequent can be simplified, using $\mathcal{D}_{una}$ and replacing $do(\alpha, S_0)$ with a fresh constant, to obtain a sequent $\Pi \Rightarrow \varphi$ where Π is UTC, and doesn't mention terms of sort action. As **LK** is not sorted, this sequent must be further simplified to remove the situation terms. To do so, we define two predicates $F_0(\vec{x})$ and $F_1(\vec{x})$ for each fluent $F(\vec{x}, s)$, and rewrite fluents in S_0 using F_0, and fluents in S_α with F_1. This rewriting will be denoted by $\mathcal{E}$, its inverse by $\mathcal{E}^{-1}$, and the final sequent by T.

By completeness of **LK** and the midsequent theorem, there is a deduction of T with midsequent $Q : \Gamma \Rightarrow \Delta$, such that Q is quantifier free, and T is proved from Q only using structural and quantifier rules. Let π and γ be the conjunction of Π and Γ respectively, and δ the disjunction of Γ. Then, by soundness of **LK**, $\gamma \models \delta$ and $\mathcal{E}(\pi) \models \mathcal{E}(\varphi)$, and by definition of the rules, $\mathcal{E}(\pi) \models \gamma$ and $\delta \models \mathcal{E}(\varphi)$. For example, in $\exists$: right, $\Delta \vee F(t) \models \Delta \vee \exists x F(x)$. It is important to notice that this property is not true for $\exists$: left, but this rule is not used because π is UTC.

In every structural or quantifier rule, the predicates used in the succedent of the upper sequent also appear in the succedent of the lower sequent. This means that every predicate in δ is used in $\mathcal{E}(\varphi)$. But, φ is uniform in S_α, so the only predicates mentioned in $\mathcal{E}(\varphi)$ are static, or derived from fluents and subscripted by 1. Then, the same is true for δ, whence $\mathcal{E}^{-1}(\delta)$ must be uniform in S_α.

Finally, the universal closure of $\mathcal{E}^{-1}(\delta)$ satisfies the statement of the lemma. $\square$

Proof. (Theorem 3) Let $\mathcal{D}$ be a UTC-BAT and α a ground action term. Assume that the progression of $\mathcal{D}$ w.r.t. α is feasible, and let $\mathcal{F}_\alpha = \{\varphi_1, \ldots, \varphi_m\}$ be such a progression. By definition, each φ_i is entailed by $\mathcal{D}$ and uniform in S_α, so by the previous lemma, there is a corresponding sentence φ_i^* such that $\mathcal{D} \models \varphi_i^*$ and $\varphi_i^* \models \varphi_i$.

Next we show that $\mathcal{F}_\alpha^\star = \{\varphi_1^\star, \ldots, \varphi_m^\star\}$ is a feasible progression, by proving that $\mathcal{F}_\alpha$ and $\mathcal{F}_\alpha^*$ are logically equivalent. Given that $\varphi_i^\star \models \varphi_i$ for every $i \in \{1, \ldots, m\}$, we have that $\mathcal{F}_\alpha^\star \models \varphi_i$ and, therefore, $\mathcal{F}_\alpha^\star \models \mathcal{F}_\alpha$. To prove the opposite direction, consider an arbitrary $\varphi_i^\star$. Given that $\mathcal{D} \models \varphi_i^\star$, and $\varphi_i^\star$ is uniform in $do(\alpha, S_0)$, we conclude that $\mathcal{F}_\alpha \models \varphi_i^\star$, which implies that $\mathcal{F}_\alpha \models \mathcal{F}_\alpha^\star$. □

The Complexity of Progressing UTC-BATs

So far, we have proved that the strong progression of a UTC-BAT is always FO-definable and representable as a UTC. Unfortunately, though, such a strong progression may not be computable, as it can consist of an infinite set of FO-sentences. It is therefore natural to ask whether one can at least verify if an FO-sentence is part of such a strong progression. In the following theorem, we give a positive answer to such question not only for universal FO-sentences with constants, which may be part of the progression itself, but also for a more expressive fragment of FO-sentences that may be implied by the progression. More precisely, a formula φ is said to be a $\forall^*\exists^*$-FO formula if φ is of the form $\forall\vec{x}\exists\vec{y}\,\psi$, where ψ is a quantifier-free FO-formula. Notice that such formulae include universal FO-formulae with constants as well as existential FO-formulae with constants. In turn, the latter include some fundamental classes of queries that could be naturally used to extract information from the progressed theory, such as conjunctive queries and unions of conjunctive queries.

Theorem 4. *Given a consistent UTC-BAT $\mathcal{D}$, a ground action term α and a $\forall^*\exists^*$-FO sentence φ that is uniform in $do(\alpha, S_0)$ and that does not mention any other functional symbol other than do, the problem of verifying whether φ is implied by the strong progression of $\mathcal{D}$ w.r.t. α is* CONEXPTIME*-complete. Moreover, the problem remains* CONEXPTIME*-hard even if the progression of $\mathcal{D}$ w.r.t. α is feasible and φ is restricted to be a ground relational atom.*

Before providing a sketch of the proof of Theorem 4, it is important to mention that the consistency requirement for $\mathcal{D}$ is there to show that the complexity of the problem does not arise from having to check that $\mathcal{D}_{S_0}$ is a consistent UTC. In addition, the restriction to UTC-BATs $\mathcal{D}$ and ground action terms α such that the progression of $\mathcal{D}$ w.r.t. α is feasible is imposed to demonstrate that the complexity of the problem neither arises from having to check that the progression of $\mathcal{D}$ w.r.t. α is feasible.

Proof of Theorem 4. (Sketch). We first prove membership. The problem of verifying whether φ is implied by the progression of $\mathcal{D}$ w.r.t. α is equivalent to the problem of verifying whether $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa} \models \varphi$, which in turn is equivalent to checking whether $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa} \cup \{\neg\varphi\}$ is inconsistent. Given that $\neg\varphi$ is an $\exists^*\forall^*$-FO sentence and $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa}$ is a set of universal FO-sentences, the problem of verifying whether $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa} \cup \{\neg\varphi\}$ is inconsistent can be reduced to deciding whether an $\exists^*\forall^*$-FO sentence is inconsistent, that is, whether an FO-sentence in the Bernays–Schönfinkel class is inconsistent (recall that no function symbol occurs in $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa} \cup \{\neg\varphi\}$). Thus, given that this latter problem is in CONEXPTIME (Lewis 1980), we conclude the membership part of the theorem.

To prove the hardness part, we show that for every nondeterministic Turing machine M that accepts a language in NEXPTIME, and for every input string w, it is possible to construct in polynomial time an instance (D, A, φ), where $\mathcal{D}$ is a UTC-BAT, A is a ground action term, and φ is a ground relational atom uniform in $do(A, S_0)$, such that the progression of $\mathcal{D}$ w.r.t. A implies φ if and only if M rejects w. More precisely, we encode in $\mathcal{D}_{S_0}$ the execution of the Turing machine M with input w. We follow an idea similar to the proof of Theorem 2, defining a fluent E_q such that $E_q(\vec{x}, S_0)$ is satisfied when the machine is in state q at time step $\vec{x}$. Different from the proof of Theorem 2, $\vec{x}$ is a tuple of n variables, each of which can take values in $\{0, 1\}$. Note that with n variables we can represent 2^n time steps, and that we can choose the value of n depending on w and M.

To define the contents of the tape and the position of the head, we use fluents T_a and H (as in the proof of Theorem 2). Our definitions for T_a and H use in turn fluent N, which defines a successor relation between time steps. The definition of N using a universal formula with constants is straightforward, given that time steps are tuples of values 0 and 1 of length n, where n depends on M and w. Finally, $\mathcal{D}_{ssa}$ contains a successor state axiom $E_{q_f}(\vec{x}, do(a, s)) \equiv a = A \wedge \neg E_{q_f}(\vec{x}, s)$, where q_f is the accepting state of M. Furthermore, for every state q that is not q_f, $\mathcal{D}_{ssa}$ contains an axiom $E_q(\vec{x}, do(a, s)) \equiv E_q(\vec{x}, s) \wedge a \neq A$. Analogously, we define the successor state axioms for other fluents such that their extensions in $do(A, S_0)$ are empty.

It only remains to show how FO-sentence φ is defined. Assuming that $\vec{c}$ is the tuple $(1, \dots, 1)$ with n values 1, we have that $\varphi = E_{q_f}(\vec{c}, do(A, S_0))$. It is possible to prove that the progression of $\mathcal{D}$ w.r.t. A is feasible, so satisfying one of the restriction imposed in the theorem. Besides, φ is a ground relational atom, so satisfying the other restriction imposed in the theorem. Thus, it only remains to prove that φ is implied by the progression of $\mathcal{D}$ w.r.t. A if and only if M does not accept w.

To see why is this the case, assume that M does not accept w. Then in every run of M over w, it holds that the state of M at time $\vec{c}$ is different from q_f. Thus, in every model $\mathbb{M}$ of $\mathcal{D}$, it holds that $\mathbb{M} \models \neg E_{q_f}(\vec{c}, S_0)$, from which we deduce that $\mathbb{M} \models E_{q_f}(\vec{c}, do(A, S_0))$ since $\mathbb{M} \models \mathcal{D}_{ssa}$. We conclude then that φ is implied by the progression of $\mathcal{D}$ w.r.t. A. For the opposite direction, assume M accepts w. Hence, there exists a run of M over w such that the state of M is q_f at time $\vec{c}$. Let $\mathbb{M}$ be a model of $\mathcal{D}$ encoding such a run, so that $\mathbb{M} \models E_{q_f}(\vec{c}, S_0)$. Then, $\mathbb{M} \models \neg E_{q_f}(\vec{c}, do(A, S_0))$ since $\mathbb{M} \models \mathcal{D}_{ssa}$, from which we conclude that φ is not implied by the progression of $\mathcal{D}$ w.r.t. A. □

When the progression of a UTC-BAT $\mathcal{D}$ w.r.t. a ground action term α is feasible, the next natural question is *how small this progression could be*. In fact, such a progression can be represented by different UTCs, as long as they are logical equivalent, so we ask whether there is a *compact* representation of such a progression as a UTC. Even more, it may be possible that there exists a fixed polynomial $p(x)$ such that if the progression of a UTC-BAT $\mathcal{D}$ w.r.t. a ground action term α is feasible, then there exists a UTC $\mathcal{F}_\alpha$ such that $\mathcal{F}_\alpha$ is a feasible progression of $\mathcal{D}$ w.r.t. α and $||\mathcal{F}_\alpha|| \leq p(||\mathcal{D}|| + ||\alpha||)$, where $||\Gamma||$ denotes the size of a set of formulae Γ (that is, the length of a string encoding Γ over an appropriate alphabet). Notice that Theorem 4 per se does not preclude such a possibility. Nevertheless, in the following result we show that there exists a family of UTC-BATs $\mathcal{D}^2, \mathcal{D}^3, \mathcal{D}^4, \ldots$, such that the progression of each $\mathcal{D}^n$ w.r.t. α is feasible but the size of every feasible progression of $\mathcal{D}^n$ w.r.t. α grows exponentially with n. (Recall $\mathcal{D}^n_{S_0}$ and $\mathcal{D}^n_{ssa}$ refer to the initial situation description and the set of successor state axioms of theory $\mathcal{D}^n$, respectively.)

Theorem 5. *There exists a family of UTC-BATs $\{\mathcal{D}^n\}_{n\geq 2}$ and a ground action term α such that the progression of $\mathcal{D}^n$ w.r.t. α is feasible, and the size of every representation of such a progression as a UTC has size $\Omega(2^{||\mathcal{D}^n_{S_0}||+||\mathcal{D}^n_{ssa}||})$.*

Proof. (Sketch.) For $n \geq 2$, the UTC-BAT $\mathcal{D}^n$ is defined over a vocabulary consisting of constants $a_1, \ldots, a_n$ and two fluents L and T. Following the proof for Corollary 1, $\mathcal{D}^n_{S_0}$ contains axioms expressing that L defines a (strict) linear order, but in this case between pairs of element of the domain in S_0. For example, the following axioms indicate that L is irreflexive and transitive:

$$\forall x\, \neg L(x, x, x, x, S_0),$$

$$\forall x_1 \forall x_2 \forall y_1 \forall y_2 \forall z_1 \forall z_2\, [(L(x_1, x_2, y_1, y_2, S_0) \wedge L(y_1, y_2, z_1, z_2, S_0)) \supset L(x_1, x_2, z_1, z_2, S_0)].$$

In addition, $\mathcal{D}^n_{S_0}$ contains an axiom stating that L is only defined for the n constants $a_1, \ldots, a_n$:

$$\forall x_1 \forall x_2 \forall y_1 \forall y_2 \left[L(x_1, x_2, y_1, y_2, S_0) \supset \left(\left(\bigvee_{i=1}^{n} x_1 = a_i \right) \wedge \left(\bigvee_{i=1}^{n} x_2 = a_i \right) \wedge \left(\bigvee_{i=1}^{n} y_1 = a_i \right) \wedge \left(\bigvee_{i=1}^{n} y_2 = a_i \right) \right) \right].$$

Fluent T has arity n^2+1, it is empty in S_0, and it contains in $do(A, S_0)$ all elements in the liner order L for the n^2 pairs. Specifically, $\mathcal{D}_{ssa}$ contains the following axioms:

$$L(\vec{x}_1, \vec{x}_2, do(a, s)) \equiv L(\vec{x}_1, \vec{x}_2, s) \wedge a \neq A$$

$$T(\vec{x}_1, \ldots, \vec{x}_{n^2}, do(a, s)) \equiv \left(a = A \wedge \bigwedge_{i=1}^{n^2-1} L(\vec{x}_i, \vec{x}_{i+1}, s) \right) \vee T(\vec{x}_1, \ldots, \vec{x}_{n^2}, s),$$

where $\vec{x}_i$ stands for the two variables $x_{1,i}, x_{2,i}$.

Finally, $\mathcal{D}^n_{S_0}$ also expresses that a_i is different from a_j if $i \neq j$. Note that the sizes of $\mathcal{D}^n_{S_0}$ and $\mathcal{D}_{ssa}$ are both quadratic in n, and so it is the size of $\mathcal{D}^n$. Now the progression of $\mathcal{D}^n$ w.r.t. A can be expressed as a universal FO-sentence with constants, which is a conjunction of two sentences: the first one says that L is empty in $do(A, S_0)$, and the second one is a disjunction of formulae expressing that T in $do(A, S_0)$ must contain all $n^2!$ permutations of the n^2 pairs of constants $a_1, \ldots, a_n$. It is possible to show that a smaller universal formula with constants would not represent a correct progression. Finally, the result follows from the fact that $n^2!$ is $\Omega(2^{n^2})$ and the size of $\mathcal{D}^n_{S_0} \cup \mathcal{D}^n_{ssa}$ is quadratic in n. $\square$

Resolution-Based Progression of UTC-BATs

We showed that progressing a UTC-BAT always yields a UTC-BAT (Theorem 1). While this tells us how the progression of a UTC-BAT looks like, it does not tell us *how* to compute it. The goal of this section is to give an answer to this question by focusing on finite UTC-BATs, since these are the only ones that we can hope to progress in practice.

Since the progression of a UTC-BAT could be infinite (Corollary 1), it is not possible in general to compute it. We show below, though, that a progression of a UTC-BAT w.r.t. some action term can be *enumerated* using a resolution-based procedure. For some input theories, our procedure terminates. In such a case, as we prove below, its output is a UTC that is a strong progression. In the rest of the section, we exploit first-order resolution (Robinson 1965). We introduce some background below.

FO Resolution and Its Enumeration

A first-order *clause* is a set $\{\ell_1(\vec{t}_1), \ell_2(\vec{t}_2), \ldots, \ell_n(\vec{t}_n)\}$, where, for every $i \in \{1, \ldots, n\}$, $\vec{t}_i$ is a sequence of terms, and $\ell_i(\vec{t}_i)$ is a first-order *literal*, that is, either an atomic formula or the negation of an atomic formula. Intuitively, a clause C represents the disjunction $\forall(C) = \forall \vec{x} \bigvee_{\ell \in C} \ell$, where $\vec{x}$ contains all variables that are free in some $\ell \in C$. Given a literal r, its *complement* (i.e., the smallest formula equivalent to $\neg r$) is denoted by $\bar{r}$ (if $r = P(\vec{t})$, then $\bar{r} = \neg P(\vec{t})$; and if $r = \neg P(\vec{t})$, then $\bar{r} = P(\vec{t})$). A *substitution* is a partial mapping of variables to terms. If r is a literal, $r\theta$ denotes the literal that results from substituting every occurrence of x in C by $\theta(x)$, for every x that is mapped in θ. If C is a clause, then $C\theta$ is the clause that results by substituting every literal in C with θ. Likewise, if $\mathcal{C}$ is a set of clauses $\mathcal{C}\theta = \{C\theta \mid C \in \mathcal{C}\}$. A substitution θ is a *unifier* of two literals ℓ and ℓ' if $\ell\theta = \ell'\theta$. A unifier θ is a most general unifier (MGU) of ℓ and ℓ' if for every unifier θ' of ℓ and ℓ' there exists a substitution ρ such that $\theta' = \theta \circ \rho$, where $\circ$ denotes the composition of functions.

The *resolution of* C_1 *and* C_2, denoted by $Res(C_1, C_2)$, is a set obtained with the following procedure. We start making $Res(C_1, C_2)$ equal to $\emptyset$. Then, for each $\ell_1 \in C_1$ and $\ell_2 \in C_2$ such that a unifier ρ for ℓ_1 and $\bar{\ell}_2$ exists, we add to $Res(C_1, C_2)$ the clauses of form $(C_1 \cup C_2)\theta \setminus \{\ell_1, \ell_2\}\theta$, where θ is a MGU of ℓ_1 and $\bar{\ell}_2$. Each formula in $Res(C_1, C_2)$ is entailed by $\{C_1, C_2\}$ (Robinson 1965).

We say that a clause C can be proven by resolution from $\mathcal{C}$, denoted $\mathcal{C} \vdash C$, *if and only if* $C \in \mathcal{C}$ or there exists two clauses C_1 and C_2 which can be each proven by resolution from $\mathcal{C}$ such that $C \in Res(C_1, C_2)$. When $\mathcal{C} \vdash C$, there exists at least a *proof tree* for C, which is a binary tree where each node is labeled by a clause, the root is labeled by C, and the leaves are labeled with nodes in $\mathcal{C}$. Finally, if a non-leaf node is labeled by D and its children are labeled by C_1 and C_2, then $D \in Res(C_1, C_2)$. If $\mathcal{C} \vdash C$, we say that d is the proof-depth of C if and only if d is the height of a shortest proof tree for C. To handle languages with equality, it is necessary to augment the set $\mathcal{C}$ with the clausal version of the *axioms of equality*, denoted by *Eq* (these are standard axioms that we omit to save space).

Algorithm 1: Enumerating Resolution Algorithm

1 **procedure** ENUMRESOLUTION($\mathcal{C}$)
2   **for each** $C \in \mathcal{C}$ **do** PRINT(C)
3   $L_0 \leftarrow \mathcal{C} \cup Eq$
4   **for** $d \leftarrow 0 \ldots \infty$ **do**
5   *Next* $\leftarrow \emptyset$
6   **foreach** $C_0 \in L_d$ *and* $C_1 \in \cup_{i=0}^{d} L_i$ **do**
7       $S \leftarrow Res(C_0, C_1) \setminus \cup_{i=1}^{d} L_i$
8   **foreach** $C \in S$ **do** PRINT(C)
9   *Next* $\leftarrow$ *Next* $\cup\, S$
10   **if** *Next* $= \emptyset$ **then return**
11     $L_d \leftarrow$ *Next*

For a finite set of first-order clauses $\mathcal{C}$, procedure ENUMRESOLUTION, shown in Algorithm 1, will enumerate every clause C such that $\mathcal{C} \vdash C$. Intuitively, in the d-th iteration of the main loop (Lines 4–11), all clauses of proof-depth d are printed via the procedure PRINT.

Resolution is refutation-complete, that is, if $\mathcal{C}$ does not have a model, the empty clause will be eventually printed by ENUMRESOLUTION. However, if $\mathcal{C} \models \forall(C)$, there is no guarantee that C appears in the enumeration. However, the following result establishes that an enumeration of $\mathcal{C}$ mentions the simplest non-tautological clauses that are entailed by $\mathcal{C}$. In this result, C' subsumes C means that there exists a substitution θ under which $C'\theta \subseteq C\theta$.

Theorem 6 (Subsumption theorem (Lee 1967))**.** *Let Σ be a set of clauses and C a non-tautological clause. Then, $\forall(\Sigma) \models \forall(C)$ iff $\Sigma \vdash C'$ and C' subsumes C.*

Progression via Enumeration

Now we are ready to present a resolution-based progression method. Recall that from Theorem 1, a strong progression of a UTC-BAT $\mathcal{D}$ w.r.t. α can be represented as a (first-order) UTC theory. An important consequence of this, that we exploit below, is that a weak progression is also a strong progression of the theory. This can be proven using the same argument we used to prove Proposition 1.

So, let us prove that Algorithm 1 can be used to enumerate a strong progression. First, recall that a weak progression is a set of FO-sentences uniform in $do(\alpha, S_0)$ that entails every formula uniform in $do(\alpha, S_0)$ which in turn is entailed by the original theory $\mathcal{D}$. As a consequence of Reiter (1991)'s regression theorem, every formula uniform in $do(\alpha, S_0)$ entailed by $\mathcal{D}$ is entailed by $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa}[\alpha, S_0] \cup \mathcal{D}_{una}$, where $\mathcal{D}_{ssa}[\alpha, S_0]$ is the set of FO-sentences that results of substituting a with α and s with S_0 in every successor state axiom $F(\vec{x}, do(a, s)) \equiv \Phi_F(\vec{x}, a, s)$ in $\mathcal{D}_{ssa}$. Such an observation is a key element of the proof of the following theorem.

Theorem 7. *Let $\mathcal{D}$ be a universal UTC-BAT and α a ground action term. The FO-sentences corresponding to the clauses enumerated by* ENUMRESOLUTION($\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa}[\alpha, S_0] \cup \mathcal{D}_{una}$) *that are uniform in $do(\alpha, S_0)$ is an enumeration of the strong progression of $\mathcal{D}$ w.r.t. α.*

Proof. Form Theorem 6 and the observations above. □

When the enumeration of resolution terminates, we obtain a feasible progression of the UTC-BAT.

Corollary 2. *Let $\mathcal{D}$ be a UTC-BAT and α a ground action term. If the call* ENUMRESOLUTION($\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa}[\alpha, S_0] \cup \mathcal{D}_{una}$) *terminates, then the set of formulae uniform in $do(\alpha, S_0)$ corresponding to the clauses printed by the algorithm is a strong (and feasible) progression of $\mathcal{D}$ w.r.t. α.*

Example 1. *Consider a theory $\mathcal{D}$ with $\mathcal{D}_{ssa}$ defined by axioms* (2) *and* (3), *$\mathcal{D}_{S_0} = \{Glass(x) \supset (Holding(x, S_0) \vee Broken(x, S_0))\}$, and $\mathcal{D}_{una}$ stating that actions dropAll and pick are different. Let $\alpha = dropAll$ and $S_\alpha = do(\alpha, S_0)$. Below we show the clauses that correspond to $\mathcal{D}_{S_0} \cup \mathcal{D}_{ssa}[\alpha, S_0] \cup \mathcal{D}_{una}$ after simplification with the axioms in $\mathcal{D}_{una}$ and elimination of subsumed clauses:*

$$\neg Glass(x) \vee Holding(x, S_0) \vee Broken(x, S_0) \tag{4}$$

$$Broken(x, S_\alpha) \vee \neg Broken(x, S_0) \tag{5}$$

$$Broken(x, S_\alpha) \vee \neg Holding(x, S_0) \tag{6}$$

$$\neg Broken(x, S_\alpha) \vee Holding(x, S_0) \vee Broken(x, S_0) \tag{7}$$

$$\neg Holding(x, S_\alpha) \tag{8}$$

By successive resolution, Algorithm 1 prints two formulae uniform in S_α. Specifically, $\neg Glass(x) \vee Broken(x, S_\alpha)$, by resolving (4) *with* (5) *and* (6)*, and $\neg Holding(x, S_\alpha)$. It is straightforward to verify that the enumeration algorithm terminates, and thus it has computed a feasible progression.*

Summary

We studied the problem of progressing the knowledge base of an agent after an action has been performed under the syntactically (and conceptually) simple fragment of well-known Basic Action Theories (Reiter 2001): universal formulae with constants, itself a special case in *Bernays-Schönfinkel class* (Bernays and Schönfinkel 1928). We show first that the progression of a UTC-BAT is always definable in first-order logic and that it can be always represented by a (possibly infinite) UTC. We then demonstrated that determining whether the progression is feasible is, however, an undecidable problem. In our analysis, we put an emphasis on computational complexity, showing that deciding whether a formula is entailed by a progression of a UTC-BAT is a computationally difficult problem; indeed, it is CONEXPTIME-complete. Finally, we showed that the progression of a UTC-BAT can always be enumerated, and sometimes effectively computed, by resorting to the standard resolution.

References

Bernays, P., and Schönfinkel, M. 1928. Zum Entscheidungsproblem der mathematischen Logik. *Mathematische Annalen* 99(1):342–372.

De Giacomo, G.; Lespérance, Y.; Levesque, H. J.; and Sardina, S. 2009. IndiGolog: A high-level programming language for embedded reasoning agents. In Bordini, R. H.; Dastani, M.; Dix, J.; and Fallah-Seghrouchni, A. E., eds., *Multi-Agent Programming: Languages, Platforms and Applications*. Springer. chapter 2, 31–72.

De Giacomo, G.; Lespérance, Y.; and Levesque, H. J. 2000. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence* 121(1–2):109–169.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Gelfond, M., and Lifschitz, V. 1998. Action languages. *Electronic Transactions of AI (ETAI)* 2:193–210.

Ghallab, M.; Nau, D. S.; and Traverso, P. 2014. The actor's view of automated planning and acting: A position paper. *Artificial Intelligence* 208:1–17.

Hodges, W. 1997. *A Shorter Model Theory*. Cambridge University Press.

Kowalski, R. A., and Sergot, M. J. 1986. A logic-based calculus of events. *New Generation Computing* 4:67–95.

Kreisel, G., and Krivine, J. 1976. *Elements of mathematical logic. Model theory*. Amsterdam: North-Holland.

Lakemeyer, G., and Levesque, H. J. 2002. Evaluation-based reasoning with disjunctive information in first-order knowledge bases. In *KR*, 73–81.

Lee, C.-T. 1967. A completeness theorem and computer program for finding theorems derivable from given axioms. *Ph. D. thesis, Department of Electrical Engineering and Computer Science, University of California*.

Levesque, H. J., and Lakemeyer, G. 2001. *The Logic of Knowledge Bases*. The MIT Press.

Levesque, H. J.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. B. 1997. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming* 31:59–84.

Lewis, H. R. 1980. Complexity results for classes of quantificational formulas. *Journal of Computer and System Sciences* 21(3):317–353.

Libkin, L. 2004. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer.

Lin, F., and Reiter, R. 1994. How to progress a database (and why) I. logical foundations. In *KR*, 425–436.

Lin, F., and Reiter, R. 1995. How to progress a database II: The STRIPS connection. In *IJCAI*, 2001–2007.

Lin, F., and Reiter, R. 1997. How to progress a database. *Artificial Intelligence* 92:131–167.

Liu, Y., and Lakemeyer, G. 2009. On first-order definability and computability of progression for local-effect actions and beyond. In *IJCAI*, 860–866.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence* 4:463–502.

McIlraith, S. A. 2000. Integrating actions and state constraints: A closed-form solution to the ramification problem (sometimes). *Artificial Intelligence* 116(1-2):87–121.

Pinto, J. A. 1994. *Temporal Reasoning in the Situation Calculus*. Ph.D. Dissertation, Department of Computer Science, University of Toronto, Toronto, Canada.

Pinto, J. 1999. Compiling ramification constraints into effect axioms. *Computational Intelligence* 15:280–307.

Pirri, F., and Reiter, R. 1999. Some contributions to the metatheory of the situation calculus. *Journal of the ACM* 46(3):261–325.

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*. Academic Press. 359–380.

Reiter, R. 1993. Proving properties of states in the situation calculus. *Artificial Intelligence* 64:337–351.

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In *KR*, 2–13.

Reiter, R. 2001. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press.

Robinson, J. A. 1965. A machine-oriented logic based on the resolution principle. *Journal of the ACM* 12:23–41.

Sandewall, E. 1994. *Features and Fluents. A Systematic Approach to the Representation of Knowledge about Dynamical Systems*, volume 1. Oxford University Press.

Sardina, S.; De Giacomo, G.; Lespérance, Y.; and Levesque, H. J. 2004. On the semantics of deliberation in IndiGolog – From theory to implementation. *Annals of Mathematics and Artificial Intelligence* 41(2–4):259–299.

Shanahan, M. 1997. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. London, England: The MIT Press.

Shanahan, M. 1999. The event calculus explained. In Levesque, H., and Pirri, F., eds., *Artificial Intelligence Today*, volume 1600 of *Lecture Notes in Computer Science (LNCS)*. Springer. 409–430.

Takeuti, G. 1987. *Proof theory*. Number 81 in Studies in logic and the foundations of mathematics. North-Holland, 2nd edition.

Thielscher, M. 1999. From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence* 111(1-2):277–299.

Vassos, S., and Levesque, H. 2008. On the progression of situation calculus basic action theories: Resolving a 10-year-old conjecture. In *AAAI*, 1004–1009.

Vassos, S., and Levesque, H. J. 2013. How to progress a database III. *Artificial Intelligence* 195:203–221.

Vassos, S., and Sardina, S. 2011. A database-type approach for progressing action theories with bounded effects. In Lakemeyer, G., and McIlraith, S. A., eds., *Knowing, Reasoning, and Acting: Essays in Honour of Hector J. Levesque*. College Publications. chapter 29, 467–486.

Vassos, S.; Lakemeyer, G.; and Levesque, H. 2008. First-order strong progression for local-effect basic action theories. In *KR*, 662–672.