On Progression and Query Evaluation in First-Order Knowledge Bases with Function Symbols

Vaishak Belle and **Gerhard Lakemeyer**
Dept. of Computer Science
RWTH Aachen
52056 Aachen
Germany
{belle,gerhard}@cs.rwth-aachen.de

Abstract

In a seminal paper, Lin and Reiter introduced the notion of progression of basic action theories. Unfortunately, progression is second-order in general. Recently, Liu and Lakemeyer improve on earlier results and show that for the local-effect and normal actions case, progression is computable but may lead to an exponential blow-up. Nevertheless, they show that for certain kinds of expressive first-order knowledge bases with disjunctive information, called proper$^+$, it is efficient. However, answering queries about the resulting state is still undecidable. In this paper, we continue this line of research and extend proper$^+$ KBs to include functions. We prove that their progression wrt local-effect, normal actions, and range-restricted theories, is first-order definable and efficiently computable. We then provide a new logically sound and complete decision procedure for certain kinds of queries.

Introduction

A fundamental problem in reasoning about action is *projection*, which is to determine if a formula holds after a number of named actions have occurred, given a logical axiomatization of how the world behaves. In the situation calculus [Reiter, 2001], a *regression* operator solves the problem by reducing entailments about the future to a query about the initial knowledge base (KB). However, it is generally agreed that the use of regression is only reasonable on a small number of actions. In a seminal paper, Lin and Reiter (LR) [1997] developed the theory of *progression*, where the idea is to update the initial KB. There are two clear advantages with progression: no duplication of effort is needed to answer multiple queries about the resulting state, and second, one imagines that an agent, during its idle time, can compute progression while doing other physical activities.

In practice, progression has three main computational requirements: the new KB must be efficiently computable, its size must at most be linear in the size of initial KB (to allow progression to iterate), and lastly, the query evaluation problem must at least be decidable. Unfortunately, LR's definition comes at the cost of second-order (SO) sentences in the progressed KB. And even if it is first-order (FO), the new theory may be an infinite one. Recently, Liu and Lakemeyer [2009] improve on earlier results and show that for a large class of action theories, called local-effect and normal actions, progression is FO definable and computable, but may lead to an exponential blow-up. Nevertheless, they also show that for certain kinds of FO disjunctive information, called *proper$^+$KBs*, progression is efficiently computable. Briefly, since databases, which are equivalent (under certain assumptions) to a maximally consistent set of function-free ground *literals*, are too restrictive for KR purposes, proper$^+$KBs were introduced [Lakemeyer and Levesque, 2002], which generalize databases and are equivalent to a (possibly) infinite set of consistent (not necessarily maximal) function-free ground *clauses*. For example, one can include things like $\mathit{Graduate}(\mathit{john}) \lor \mathit{Graduate}(\mathit{mary})$ or $\forall x.(\mathit{Graduate}(x) \supset \mathit{Student}(x))$. So while the first two requirements of progression are accounted for (at least under some restrictions) the third, unfortunately, is not easy to satisfy. Unrestricted first-order initial theories is clearly asking too much, but deductive reasoning with proper$^+$KBs, which can be seen as a *language* restriction via syntactic normalization, remains undecidable.

There is a well-known tradeoff between the expressiveness of the representation language and its computational behavior. Over the decades, two main techniques have emerged. In the first, small domain FO theories are translated into propositional ones, often augmented with domain dependent information [Kautz and Selman, 1992]. In the second, FO features are not compromised but entailment is *weakened*, by way of non-traditional semantics *e.g.* [Liu *et al.*, 2004][1]. That said, both methods have limited appeal as they are function-free. This does not always coincide with practice, where in many standard applications such as moving robots [Levesque and Lakemeyer, 2001], game playing agents and planners [Reiter, 2001], functions are either essential or in the least allow for succinct representations. The apparent difficulty is that even simple clauses such as $\mathit{grade} = 4 \lor \mathit{grade} \neq 2$ result in non-trivial encodings: first, there is the semantic property that ground functions obtain unique values, and second, $\mathit{grade} \neq 2$ says that there are an infinite number of possible values other than 2.

In this paper, we continue this line of research, where we investigate cases of progression that remain practical yet expressive. We extend proper$^+$KBs to include functions, and show that progression wrt local-effect and normal actions is

[1] Liu *et al.* propose a decidable but *incomplete* reasoning procedure for (function-free) proper$^+$KBs. However, the conditions under which reasoning becomes complete is left open.

FO definable and efficiently computable. But they do not cover range-restricted actions [Vassos *et al.*, 2009] which include non-local actions such as an exploding bomb that destroys everything in the vicinity. We obtain computability results for this class as well. Then for a large class of FO queries we provide a new methodology for sound and complete reasoning; one that is inspired by Boolean satisfiability.

In the next section, we introduce the formalism, and then present results in the order indicated. For space reasons, the paper contains no proofs. They are presented in [Belle, 2011].

The Logic $\mathcal{ES}_O$

We consider a modal reconstruction of the situation calculus called $\mathcal{ES}_O$, with epistemic features, including *only-knowing* [Levesque and Lakemeyer, 2001], which refers to all that an agent knows in the sense of having a KB. In recent work, Lakemeyer and Levesque [2009] show how the semantics below is fully compatible with LR's idea of progression. So what we obtain in this paper is a definition of FO progression in an epistemic setting with regards to actions with non-trivial sensing results, and where one can analyze beliefs and non-beliefs as valid sentences.[2] We review the main features:

1. *Language*: It includes fluent functions and rigid functions (for *actions* only) of every arity,[3] rigid SO functions of every arity, FO variables, SO function variables, distinguished fluents *Poss* and *SF* (for sensing), and closed under connectives: $\wedge, \neg, =, \forall, [t], \Box, \boldsymbol{B}, \boldsymbol{O}$.[4] We assume there is a countably infinite set of *object* names $\mathcal{N}$; *e.g. obj5, desk*. Let $\mathcal{A}$ be the (infinite) set $\{A(m_1, \ldots, m_k) \mid A \text{ is a rigid function}, m_j \in \mathcal{N}\}$, and these denote *action* names; *e.g. move(obj5, desk)*. $\mathcal{N} \cup \mathcal{A}$ serves as the domain of discourse.[5] We also assume that variables come in both the *object* and the *action* sort.
2. *Terms*: Every FO variable and name is a term. We use $\boldsymbol{t}$ to denote a vector of terms, and t_j to denote a term in $\boldsymbol{t}$. If f is a function and R is a SO function variable, then $f(\boldsymbol{t})$ and $R(\boldsymbol{t})$ are terms. By *primitive term* and *primitive SO term*, we mean ones of the form $f(\boldsymbol{m})$ and $R(\boldsymbol{m})$ resp. , where $m_j \in \mathcal{N}$.
3. *Formulas*: Let t and t' be terms. If α and β are formulas, then so are: $t = t', \alpha \wedge \beta, \neg\alpha, \forall x\alpha, \forall R\alpha, [t]\alpha, \boldsymbol{B}\alpha, \boldsymbol{O}\alpha, \Box\alpha$. *Primitive equalities* are formulas of the form $f(\boldsymbol{m}) = n$, where $m_j, n \in \mathcal{N}$. *Fluent literals* are of the form $f(\boldsymbol{r}) = s$, where f is neither *Poss* nor *SF*, and r_j and s are either variables or names. A *clause* is a disjunction of such literals. *Fluent formulas* are those that only mention fluent literals.[6]

[2] While our results also apply to the non-epistemic fragment, we believe the ability to deal with sensing is an important feature. See [Lakemeyer and Levesque, 2009] for examples.

[3] Predicates are not included for simplicity, and can be thought of as special functions whose values are either (say) name 0 or name 1.

[4] Symbols such as $\vee, \exists, \equiv$ and $\supset$ are understood as usual.

[5] This allows a substitutional interpretation of quantifiers (respecting sorts).

[6] Formulas with nested functions can be expressed wlog as a fluent formula. For instance, $f(g(x)) = h(x)$ is written equivalently as $\exists z, z'\ f(z) = z' \wedge g(x) = z \wedge h(x) = z'$.

We let $\mathcal{Z}$ denote all finite sequences of names in $\mathcal{A}$, including $\langle\rangle$ *i.e.*, empty sequence. We define *worlds* as functions from (fluent) primitive terms and $\mathcal{Z}$ to $\mathcal{N}$, and from primitive SO terms to $\mathcal{N}$. An *epistemic state* $e \subseteq \mathcal{W}$ is any set of worlds. While names act as rigid designators, the co-referring name for an arbitrary term is obtained wrt w and $z \in \mathcal{Z}$ as:
(a) $|t|^z_w = t$ if $t \in \mathcal{N} \cup \mathcal{A}$;
(b) $|A(\boldsymbol{t})|^z_w = A(\boldsymbol{m})$, where $m_j = |t_j|^z_w$ and A is an action function;
(c) $|f(\boldsymbol{t})|^z_w = w[f(\boldsymbol{m}), z]$, where f is a fluent, $m_j = |t_j|^z_w$;
(d) $|R(\boldsymbol{t})|^z_w = w[R(\boldsymbol{m})]$, where $m_j = |t_j|^z_w$.

For sensing, we consider a relation $w' \simeq_z w$ s.t. $w' \simeq_{\langle\rangle} w$ for all w' and w, and $w' \simeq_{z \cdot t} w$ iff $w' \simeq_z w$, $w'[\mathit{Poss}(t), z] = 1$ and $w'[\mathit{SF}(t), z] = w[\mathit{SF}(t), z]$.

To interpret SO variables, we write $w' \sim_R w$ to mean that w' and w agree on everything except assignments involving R. Now, given e, w, z, a semantics is (connectives are understood as usual):

- $e, w, z \models t_1 = t_2$ iff n_1 and n_2 are identical, $n_j = |t_j|^z_w$;
- $e, w, z \models [t]\alpha$ iff $e, w, z \cdot \sigma \models \alpha$, where $\sigma = |t|^z_w$;
- $e, w, z \models \Box\alpha$ iff $e, w, z \cdot z' \models \alpha$ for all $z' \in \mathcal{Z}$;
- $e, w, z \models \forall x\alpha$ iff $e, w, z \models \alpha^x_n$ for every name n of the right sort;
- $e, w, z \models \forall R\alpha$ iff $e, w', z \models \alpha$ for every $w' \sim_R w$;

A new feature of $\mathcal{ES}_O$ is that belief is handled by *progressing* epistemic states wrt actions. That is, let w_z be a world such that $w_z[p, z'] = w[p, z \cdot z']$ for all primitive terms p and action sequences z'. Further, let $e^w_z = \{w'_z \mid w' \in e \text{ and } w' \simeq_z w\}$.

- $e, w, z \models \boldsymbol{B}\alpha$ iff for all $w' \in e^w_z$, $e^w_z, w', \langle\rangle \models \alpha$;
- $e, w, z \models \boldsymbol{O}\alpha$ iff for all w', $w' \in e^w_z$ iff $e^w_z, w', \langle\rangle \models \alpha$.

Models satisfy well-known introspective properties, *i.e.* that of weak **S5** [Hughes and Cresswell, 1972; Lakemeyer and Levesque, 2009]. Given a set of sentences Σ, we write $\Sigma \models \alpha$ to mean that for every e, w, z if $e, w, z \models \alpha'$ for all $\alpha' \in \Sigma$ then $e, w, z \models \alpha$. Finally, $\models \alpha$ denotes $\{\} \models \alpha$.

The Computability of Progression

We begin by considering the equivalent of basic action theories (BATs) of the situation calculus [Reiter, 2001].

Definition 1: *Given a set of fluents $\boldsymbol{F}$, a set $\Sigma \subseteq \mathcal{ES}_O$ is called the basic action theory over $\boldsymbol{F}$ iff Σ is the union of:*[7]

1. *The initial theory Σ_0 is any set of fluent sentences. Σ_{pre} is a sentence of the form $\Box Poss(v) = 1 \equiv \pi$ and Σ_{sense} is a sentence of the form $\Box SF(v) = y \equiv \psi$, where π, ψ are fluent formulas and Poss represents a predicate.*
2. *Successor-state axioms (SSAs) Σ_{post} is a set of sentences of the form $\Box[v]f(\boldsymbol{x}) = y \equiv \gamma^*_f$, where $\gamma^*_f = \gamma_f(\boldsymbol{x}, y, v) \vee f(\boldsymbol{x}) = y \wedge \neg\exists h\ \gamma_f(\boldsymbol{x}, h, v)$ one for each fluent f and γ_f is a fluent formula.*

Denote the initial theory as ϕ and the rest as $\Box\beta$. Hereafter, let t denote a name from $\mathcal{A}$, say $A(\boldsymbol{e})$. We remark that the unique name assumption for actions is built into the logic.

Lakemeyer and Levesque [2009] obtain a characterization of progression in the logic in terms of only-knowing:

[7] Free variables are (implicitly) universally quantified.

Theorem 1: [Lakemeyer and Levesque, 2009] *Let t be a standard action name, then*

$$\models \boldsymbol{O}(\phi \wedge \Box\beta) \wedge SF(t) = x \supset [t]\boldsymbol{O}(\Psi \wedge \Box\beta),$$

where, Ψ *is the new initial theory defined as*

$$\exists \boldsymbol{R}.\ [(\phi \wedge \pi_t^v \wedge \psi_t^v)^{\boldsymbol{F}}_{\boldsymbol{R}} \wedge \bigwedge \forall \boldsymbol{x}, y. f(\boldsymbol{x}) = y \equiv \gamma^{*v\boldsymbol{F}}_{f\,t\,\boldsymbol{R}}]$$

s.t. $\boldsymbol{R}$ *correspond to SO function variables in that* R_j *has the same arity as* f_j*.*

We note that, different from LR, progression takes both the action executed (π_t^v) and the sensing result (ψ_t^v) into account.

In the sequel, we are interested in efficient progression, and we realize this by restricting the initial theory to so-called proper$^+$KBs (but appropriately generalized to functions) and considering certain classes of syntactically restricted BATs.

Proper$^+$KBs

Let e denote Boolean combinations of formulas of the form $r = s$, where r and s are either variables or names, and d denote clauses. Let $\forall\alpha$ denote the universal closure of α. Moreover, we call a formula of the form $\forall(e \supset d)$ a $\forall$-clause.

Definition 2: *A proper$^+$KB is any finite and satisfiable set of $\forall$-clauses.*

Restrictions on BATs

Local-effects. Actions in many domains have local-effects in the sense that if a primitive action $A(\boldsymbol{e})$ affects a fluent atom $f(\boldsymbol{m})$, then $\boldsymbol{m}$ is contained in $\boldsymbol{e}$. They generalize the *strictly context-free* class of BATs considered by LR [1997].

Definition 3: *A SSA is local-effect if $\gamma_f(\boldsymbol{x}, y, v)$ is a disjunction of formulas of the form $\exists \boldsymbol{u}\ [v = A(\boldsymbol{z}) \wedge \mu(\boldsymbol{z})]$, where $\boldsymbol{z}$ contains $\boldsymbol{x} \cup \{y\}$, $\boldsymbol{u}$ are the remaining variables in $\boldsymbol{z}$ and $\mu(\boldsymbol{z})$ is the context formula.*

Given a primitive action $A(\boldsymbol{e})$, local-effect SSAs can be simplified. That is, it can be shown that $\gamma_f(\boldsymbol{x}, y, A(\boldsymbol{e}))$ is equivalent to disjunctions of the form $\boldsymbol{x} = \boldsymbol{m} \wedge y = n \wedge \mu(\boldsymbol{e})$, where m_j and n are names mentioned in $\boldsymbol{e}$.

Normal Actions. Certain types of naturally occurring actions are not local-effect, *e.g.* moving a briefcase containing objects. Liu and Lakemeyer [2009] observe that these actions do not depend on the fluents on which they have non-local effects. That is, they have local-effects on all fluents mentioned in γ_f. For example, moving the briefcase also moves the contained objects without affecting the fluent *in*.

Definition 3 (cont.): *A primitive action $A(\boldsymbol{e})$ is said to have a local-effect on a fluent f if $\gamma_f(\boldsymbol{x}, y, A(\boldsymbol{e}))$ simplifies to a disjunction of formulas of the form $\boldsymbol{x} = \boldsymbol{m} \wedge y = n \wedge \mu(\boldsymbol{e})$. Let $LE(A) \subseteq \boldsymbol{F}$ be the functions on which $A(\boldsymbol{z})$ has local-effect. We call $A(\boldsymbol{z})$ normal if for each f, all fluents appearing $\gamma_f(\boldsymbol{x}, y, v)$ are also in $LE(A)$.*

We consider another class of actions, called *range-restricted theories*, the treatment of which is deferred to later.

Example 1: To first illustrate the idea of a proper$^+$KB, imagine the following incomplete facts about (stacks of) cards. We have cards[8] $c_1, c_2, \ldots$, stacks $s_1, s_2, \ldots$ and ranks $J, Q, \ldots$ Each card has a rank and is associated with a value.

Now let ϕ_1 be a proper$^+$KB defined as a conjunction of $rank(c_1) = J$ (rank of c_1 is J), $rank(c_2) = Q \vee rank(c_2) \neq K$, $in(c_1) = s_1$ (c_1 is inside the stack s_1) and $\forall. in(x) = s_1 \supset value(x) = 5$ (all cards in s_1 are valued at 5).

To illustrate BATs, let $\Sigma_0 = \{payoff = 0\} \cup \phi_1$, $\Sigma_{\text{pre}} = \{\Box Poss(v) = 1\}$ (for simplicity), $\Sigma_{\text{sense}} = \{\Box SF(readc1) = y \equiv value(c_1) = y\}$. Consider SSAs:

$$\Box[v] payoff = y \equiv \exists x. v = guess(x, y) \wedge value(x) = y \vee$$
$$payoff = y \wedge \neg\exists h.(v = guess(x, h) \wedge value(x) = h);$$

$$\Box[v] at(x) = y \equiv \exists u.\ v = move(u, y) \wedge (u = x \vee in(x) = u) \vee$$
$$at(x) = y \wedge \neg\exists u, h (v = move(u, h) \wedge (x = u \vee in(x) = u)).$$

That is, if the agent guesses the value of a card accurately, he receives a matching payoff. The agent may read card values (shown for c_1) and move stacks between locations. (Note that the *payoff* SSA is local-effect and *move* is a normal action.)

In this paper, we show that under the reasonable assumption that BATs are *quantifier free*, the progression of a proper$^+$KB wrt the defined classes of BATs is efficient.

Definition 4 : *A BAT is quantifier free if π_t^v, ψ_t^v and $\gamma_f(\boldsymbol{x}, y, v)_t^v$ can be simplified to quantifier free formulas.*

For instance, Example 1 is clearly quantifier free.

We begin by extending the FO definability results in [Liu and Lakemeyer, 2009] to a language with function symbols.[9] First, we show that progression of local-effect SSAs is obtained by *forgetting* [Lin and Reiter, 1994] a finite set of primitive equalities. However, different from [Liu and Lakemeyer, 2009], and perhaps of independent technical interest, we will need to handle the modal operators appearing in the instantiated SSAs. In addition, progression will be characterized as what the agent comes to only-know after the action, as an extension to Theorem 1. We then discuss computational costs. For normal actions, Liu and Lakemeyer [2009] formalize progression as a special case of Ackermann's Lemma, which is a predicate elimination result [Doherty *et al.*, 2001]. We obtain an analogous theorem for function symbols and discuss the complexity. Lastly, progression wrt range-restricted theories is also shown to be definable as forgetting a finite set of equalities. For reasons of space, we only go over the main aspects. See [Belle, 2011] for more details.

Forgetting

LR [1994] define a notion of forgetting from finite theories, which we adapt below for a language with functions.[10] Following their ideas, it can be shown that while forgetting primitive equalities is FO definable, forgetting functions is SO.

In what follows, let $\mathcal{S}$ denote a finite set of primitive equalities. We write $\mathcal{M}(\mathcal{S})$ to mean the set of all truth assignments to $\mathcal{S}$. Slightly abusing notation, given an equality ρ and function f, we write $w' \sim_\rho w$ and $w' \sim_f w$ to mean that w' and w agree on everything initially, except maybe ρ and (resp.) the values for equalities mentioning f.

[8] We assume that cards, ranks, values and stacks are names from $\mathcal{N}$.

[9] FO definability results for functional fluents are also mentioned (but not published) as an extension in [Liu and Lakemeyer, 2009]. Moreover, we differ in the ways discussed above.

[10] While their definitions are given for standard first-order logic and Tarskian models, we consider analogous notions for the $\mathcal{ES}_O$ semantical framework.

Definition 5: *Let λ denote an equality or a function. Given a fluent formula ϕ, we say any fluent formula ϕ' is the result of forgetting λ, denoted forget(ϕ, λ), if for any world w, $w \models \phi'$ iff there is a w' s.t. $w' \models \phi$ and $w \sim_\lambda w'$. Inductively define forget($\phi, \{\lambda_1, \ldots, \lambda_k\}$) as forget(forget($\phi, \lambda_1$), $\ldots, \lambda_k$).*

Definition 6: *Suppose $\theta \in \mathcal{M}(\mathcal{S})$. Let $\phi[\theta]$ denote replacing every occurrence of $f(\boldsymbol{t}) = t'$ in ϕ with:*

$$\bigvee_{j=1}^{k}(\boldsymbol{t} = \boldsymbol{m}_j \wedge (t' = n_j \wedge \mathrm{BOOL}_j \vee t' \neq n_j \wedge \neg\mathrm{BOOL}_j)) \vee (\bigwedge_{j=1}^{k} \boldsymbol{t} \neq \boldsymbol{m}_j \wedge f(\boldsymbol{t}) = t')$$

where, $f(\boldsymbol{m}_j) = n_j$ appears in $\mathcal{S}$ and is replaced by the truth assignment BOOL_j according to θ.[11]

Theorem 2: *(a) $\models$ forget($\phi, \mathcal{S}$) $\equiv \bigvee_{\theta \in \mathcal{M}(\mathcal{S})} \phi[\theta]$.*
(b) $\models$ forget(ϕ, f) $\equiv \exists R \phi^f_R$.

Progression of Local-Effect Theories

We now turn to local-effects. Suppose that we have simplified $\gamma_f(\boldsymbol{x}, y, A(\boldsymbol{e}))$ to disjunctions of $\boldsymbol{x} = \boldsymbol{m} \wedge y = n \wedge \mu(\boldsymbol{e})$ as discussed. Then define the *argument set* (wrt f) and *characteristic set* resp. as follows [Liu and Lakemeyer, 2009]:

1. $\Delta_f = \{(\boldsymbol{m}, n) | \boldsymbol{x} = \boldsymbol{m} \wedge y = n$ appears in $\gamma_f(\boldsymbol{x}, y, v)^v_t\}$
2. $\Omega = \{f(\boldsymbol{m}) = n \mid f$ is a fluent, $(\boldsymbol{m}, n) \in \Delta_f\}$.

Essentially, local-effects affect a finite number of primitive terms, which are those in Ω, conditioned on context formulas. Let us denote the instantiated SSAs $([v]f(\boldsymbol{x}) = y \equiv \gamma^*_f)^v_t$ as $\boldsymbol{F}_{\text{ssa}}$. So roughly speaking, progression is the addition of $\boldsymbol{F}_{\text{ssa}}$ to the initial theory, while forgetting the initial values of all primitive terms specified in Ω.

However, as $\boldsymbol{F}_{\text{ssa}}$ mentions modalities, we first convert it to a fluent formula. Let $\boldsymbol{G}$ be fresh functions s.t. g_j has the same arity as f_j. Now, let $\boldsymbol{F}_{\text{ssa}}[\boldsymbol{G}]$ denote $(f(\boldsymbol{x}) = y \equiv \gamma^{*\boldsymbol{F}}_{f\,\boldsymbol{G}})^v_t$, that is, we use $\boldsymbol{G}$ to characterize formulas that hold initially.

Theorem 3: *For local-effect action theories,*

$$\models \boldsymbol{O}(\phi \wedge \Box\beta) \wedge SF(t) = x \supset [t]\boldsymbol{O}(\Psi \wedge \Box\beta)$$

where Ψ = forget($(\phi \wedge \pi^v_t \wedge \psi^v_t)^{\boldsymbol{F}}_{\boldsymbol{G}} \wedge \boldsymbol{F}_{ssa}[\boldsymbol{G}], \Omega^{\boldsymbol{F}}_{\boldsymbol{G}})^{\boldsymbol{G}}_{\boldsymbol{F}}$.

Example 1 continued. If t denotes *guess*$(c_2, 3)$ then $\boldsymbol{F}_{\text{ssa}}$ is $\{value(c_2) = 3 \supset [t]payoff = 3, value(c_2) \neq 3 \supset [t]payoff = 0\}$. Thus, the progressed proper$^+$KB ϕ' is $\phi_1 \cup \{payoff = y \equiv (value(c_2) = 3 \wedge y = 3) \vee y = 0\}$. Due to incomplete information $\boldsymbol{O}(\phi' \wedge \Box\beta) \not\models \exists x \boldsymbol{B} payoff = x$.[12]

Note that Theorem 3 holds for any finite theory but it may not be efficient while the case of proper$^+$KBs is very efficient.

Theorem 4: *Forgetting an equality $f(\boldsymbol{m}) = n$ from a proper$^+$KB ϕ takes $O(h + (4^w l)^2)$ time, where h is the size of ϕ, l is the size of $\forall$-clauses in ϕ where f appears, and w is the maximum number of mentions of f in a $\forall$-clause in ϕ.*

It is reasonable to assume that l, w and Ω are $O(1)$. Then, progression is linear in the size of ϕ.

Progression of Normal Actions

When $f \in LE(A)$ for a normal action $A(\boldsymbol{z})$, then forgetting is as above. So what is really needed is a method of forgetting $f \in \boldsymbol{F} - LE(A)$.

A predicate elimination (*i.e.*, forgetting) result by Ackermann [Doherty *et al.*, 2001] states that given two formulas $\forall.P(\boldsymbol{x}) \supset \alpha(\boldsymbol{x})$ and $\forall.\beta(\boldsymbol{x})$, where α does not mention P and β only mentions P positively *i.e.* $\neg P$ does not occur in the NNF of β, eliminating P from the conjunction is equivalent to $\beta^{P(\boldsymbol{x})}_{\alpha(\boldsymbol{x})}$. An analogous result holds for functions:

Theorem 5: *Let β_i, α_j and δ denote fluent formulas not mentioning f. Forgetting f from*
$\delta \wedge \forall. \bigwedge(\beta_i(\boldsymbol{x}, y) \supset f(\boldsymbol{x}) = y) \wedge \bigwedge(f(\boldsymbol{x}) = y \supset \alpha_j(\boldsymbol{x}, y))$
is equivalent to $\delta \wedge \forall. \bigvee \beta_i(\boldsymbol{x}, y) \supset \bigwedge \alpha_j(\boldsymbol{x}, y)$.

That is, in any theory where the occurrence of f is *semi-Horn* [Doherty *et al.*, 2001], *i.e.* either as $\beta \supset f$ or $f \supset \alpha$, forgetting the function is FO definable. Suppose that functions appear in a proper$^+$KB in the semi-Horn form. Under the syntactic restrictions of normal actions, instantiated SSAs for $f \in \boldsymbol{F} - LE(A)$ can be simplified to the semi-Horn form. Therefore, forgetting f from the conjunction of the initial proper$^+$KB and the instantiated SSA is FO definable. It can be shown that computing the result is linear in the size of theory. We skip the only-knowing theorem for space reasons.

Example 1 continued. Suppose stacks are located on $desk_1$ or $desk_2$ and let $\phi_2 = \{at(s_1) = desk_1\} \cup \phi_1$. Then, progression wrt $move(s_1, desk_2)$ is ϕ_1 with the following instantiated SSAs: $\{\ y = desk_2 \wedge (x = s_1 \vee in(x) = s_1) \supset [t]at(x) = y,\ y \neq desk_2 \wedge (x = s_1 \vee in(x) = s_1) \supset \neg[t]at(x) = y\ \}$. Finally, replace all occurrences of $[t]f(\boldsymbol{x}) = y$ with $f(\boldsymbol{x}) = y$.

Progression of Range-restricted Theories

Non-local actions such as an exploding bomb destroy everything in the vicinity, *i.e.*, these actions do not specify the objects that they affect. Global effects such as these are believed to be one of the reasons why progression is SO. But when the range of nearby objects is restricted, Vassos *et al.* [2009] prove FO definability and computability results for certain types of initial theories which they refer to as a *database of possible closures (DBPC)*. Roughly speaking, a DBPC corresponds to disjunctions of maximally consistent sets of literals.

To restrict the range of nearby objects, they consider SSAs where $\gamma_f(\boldsymbol{x}, y, v)$ is a disjunction of formulas of the form $\exists \boldsymbol{u}[v = A(\boldsymbol{z}) \wedge \mu(\boldsymbol{z}, \boldsymbol{w})]$, where $\boldsymbol{z}$ contains y, $\boldsymbol{u}$ are the remaining variables in $\boldsymbol{z}$ but not in $\boldsymbol{x}$, $\boldsymbol{w}$ are variables in $\boldsymbol{x}$ but not in $\boldsymbol{z}$, and $\mu(\boldsymbol{z}, \boldsymbol{w})$ is the context formula.[13] For every primitive action $A(\boldsymbol{e})$, they further assume that the initial theory entails only a finite number of substitutions for the free variables $\boldsymbol{w}$ in the context formula $\mu(\boldsymbol{e}, \boldsymbol{w})$. It then follows that only a finite set of fluent atoms are affected after doing an action. So progression can again be formulated in terms of forgetting.

However, the account in Vassos *et al.* is very involved because progression must be also definable as a DBPC (to al-

---

[11] It is interesting to note, for example, that forgetting $f = 1$ from $f = 4$ is simply *true*, since $f = 4$ implies that $f \neq 1$. That is, forgetting primitive equalities also results in forgetting primitive terms.

[12] To see where only-knowing pays off, we can reduce formulas mentioning $\boldsymbol{B}$ to non-modal ones by means of the *representation theorem* [Levesque and Lakemeyer, 2001], which we do not go over for space reasons.

[13] Strictly speaking, Vassos *et al.* only consider predicates in their language. We present a functional fluent version of their notions in order to remain consistent with the rest of the paper.

low for iterated progression). We improve on this by proving FO definability results for proper$^+$KBs, which differs from their KBs in the sense that it can express notions like $\forall.x \neq c_3 \supset st(x) \neq broken$ which says that an infinite number of objects other than c_3 are not broken (but leaving the status of c_3 open). Second, the definition is a simple one via the concept of forgetting. Therefore, computability results from above are applicable.

We leave out the formal details, including the assumptions placed on context formulas, for space reasons (see [Belle, 2011]). We give an example instead:

Example 1 continued. Let $\phi_3 = \{\forall.near(bomb, y) = 1 \equiv y = c_1 \vee y = c_2, \forall.x \neq c_3 \supset st(x) \neq broken\} \cup \phi_1$. Include:

$$\Box[v]st(x) = y \equiv v = explode \wedge near(bomb, x) = 1 \wedge \\ y = broken \vee st(x) = y \wedge \neg\exists h.\gamma(x, h, v).$$

Then progression wrt *explode* is the removing of the $\forall$-clause mentioning *st* and the adding of $\{st(c_1) = broken, st(c_2) = broken, \forall.x \neq c_1 \wedge x \neq c_2 \wedge x \neq c_3 \supset st(x) \neq broken\}$.

Theorem 6: *Progression of a proper$^+$KB wrt local-effect, normal and range-restricted theories is FO definable, efficiently computable (under discussed assumptions), and definable as a proper$^+$KB.*

Query Evaluation

The query evaluation procedure we have in mind is a logically sound and complete decision procedure for certain classes of queries on proper$^+$KBs. To understand its basic principles, we begin with the (simpler) case where both the initial theory ϕ and query α are quantifier-free (QF) ground fluent formulas (*i.e.* no variables), and thus representable as ground clauses.

As hinted earlier, the computation mechanism is inspired by Boolean satisfiability where entailments are verified via refutation *i.e.* checking if $\mathcal{S} \doteq \phi \cup \{\neg\alpha\}$ is unsatisfiable. We essentially consider a variant of DPLL [Davis and Putnam, 1960], which first needs the notion of an *assignment*.

Definition 7: *Let θ denote a consistent set of positive primitive equalities, and $\mathcal{S}$ is as above. Let $[\mathcal{S}]_\theta$ denote:*

1. *given $f(\boldsymbol{m}) = n \in \theta$, replace every occurrence of $f(\boldsymbol{m}) = n$ in $\mathcal{S}$ with true, and every occurrence of $f(\boldsymbol{m}) \neq n$ or $f(\boldsymbol{m}) = n'$ for some $n' \neq n$ with false;*
2. *remove all clauses that contain at least one true literal, and delete all occurrences of false literals in clauses.*

Intuitively, like partial assignments in Boolean reasoners [Gomes *et al.*, 2008], $[\mathcal{S}]_\theta$ reduces $\mathcal{S}$ to a simpler formula which is satisfiable provided $\mathcal{S} \wedge \theta$ is. The satisfiability procedure works as follows. It takes as input any set of ground clauses $\mathcal{S}$ and returns either SAT with a satisfying assignment θ *i.e.* one that makes $[\mathcal{S}]_\theta$ true, or with UNSAT.

A fundamental step in the classical DPLL is that of choosing a literal from the remaining clauses and considering partial assignments that either sets the literal to true everywhere, or sets it to false. In our case, the branches are *possible assignments* to primitive terms appearing in the remaining set of clauses. However, the universe is infinite, and so not all names can be considered. Instead, we show that it is sufficient to consider names in $\mathcal{S}$, plus an (arbitrary) extra one.

Given a primitive term ρ, let $H(\mathcal{S}, \rho)$ be a set of names $\{n_1, \ldots, n_k\}$ s.t. $\rho \circ n_j$ appears in $\mathcal{S}$, where $\circ \in \{=, \neq\}$. If SAT is not returned for every assignment $\rho = n_j$, then we consider $\rho = n'$, for any $n' \in \mathcal{N} - H(\mathcal{S}, \rho)$.[14]

Proposition 1: *Let Θ denote the set $\{\rho = n_1, \ldots, \rho = n_k, \rho = n'\}$ obtained as above. Then $\mathcal{S}$ is satisfiable iff $\bigvee_{\theta\in\Theta}[\mathcal{S}]_\theta$ is satisfiable.*

Easy arguments show that a DPLL proof has $k+1$ branches, in contrast to the binary tree in the Boolean case. Therefore, the worst-case complexity is $O((k+1)^q)$, where q is the number of primitive terms in $\mathcal{S}$ and k is the size of $H(\mathcal{S}, \rho)$.

Theorem 7: *Given any set of ground clauses $\mathcal{S}$, $\mathcal{S}$ is unsatisfiable (satisfiable) iff* DPLL$(\mathcal{S}, \{\})$ *returns* UNSAT (SAT).

Algorithm 1: DPLL$(\mathcal{S}, \theta)$

Input: set of ground clauses $\mathcal{S}$ with $\theta = \{\}$ initially
Output: UNSAT, or a set θ such that $[\mathcal{S}]_\theta$ is *true*
$(\mathcal{S}, \theta) \doteq$ UNIT-PROPOGATE$(\mathcal{S}, \theta)$;
if $\mathcal{S}$ *contains false* **then**
    return UNSAT;
if $\mathcal{S}$ *has no* clauses *left* **then**
    output θ and return SAT;
$\rho \doteq$ any primitive term appearing in $\mathcal{S}$ and not mentioned in θ;
foreach $n \in H(\mathcal{S}, \rho)$ **do**
    if DPLL$([\mathcal{S}]_{\rho=n}, \theta \cup \{\rho = n\})$ = SAT **then**
        return SAT;
ret. DPLL$([\mathcal{S}]_{\rho=n'}, \theta \cup \{\rho = n'\})$ for any $n' \in \mathcal{N} - H(\mathcal{S}, \rho)$;

where, UNIT-PROPOGATE$(\mathcal{S}, \theta)$ is:
while $\mathcal{S}$ *does not contain false and has unit clause* $\rho = n$ **do**
    $\mathcal{S} \doteq [\mathcal{S}]_{\rho = n}$ and $\theta \doteq \theta \cup \{\rho = n\}$;

Beyond Ground Clauses. We extend the scope of the procedure to allow universally quantified queries wrt proper$^+$KBs. We will need one reasonable assumption. But first, we prove a result about inferring the validity of a universal by means of a finite number of substitutions. Below, if *H* denotes a set of names, let H_b^+ denote the union of *H* and *b* fresh names.

Theorem 8: *Let ϕ be any set of closed fluent formulas s.t. there is an infinite number of names not appearing in ϕ. Let α be a fluent formula with a single free variable x and let H be the set of names mentioned in ϕ and α. Then, $\phi \models \forall x\alpha$ iff $\phi \models \alpha^x_m$ for all $m \in H_1^+$.*

Suppose ϕ is a proper$^+$KB. The idea now will be to obtain a finite representation for $gnd(\phi) = \{d^{\boldsymbol{x}}_{\boldsymbol{m}} \mid \forall(e \supset d) \in \phi$ and $\models e^{\boldsymbol{x}}_{\boldsymbol{m}}\}$, which is the (possibly infinite) QF form of ϕ. We let $gnd(\phi)|H_b^+$ denote the restriction to names from H_b^+.

Theorem 9: *Let α be a closed QF fluent formula, ϕ be a proper$^+$KB, and H be as above. Let ϕ also satisfy: for every primitive term $f(\boldsymbol{m})$ with $m_j \in H$, $gnd(\phi)$ mentions only finitely many inequalities of the form $f(\boldsymbol{m}) \neq n_j$ or*

[14]To see why the fresh name is needed, let ϕ be $f(1) \neq 1$. Clearly, ϕ is satisfiable, but only for assignments to $f(1)$ other than 1. Also, for functions representing relations, the procedure is modified to only consider 0 or 1 as possible assignments (see footnote 1).

gnd$(\phi)|H_1^+$ *entails* $\bigvee f(\boldsymbol{m}) = n_j$, *where* $n_j \in H$.[15] *Then,* $\phi \models \alpha$ *iff gnd*$(\phi)|H_1^+ \models \alpha$.

Here, the "$\Leftarrow$" direction is immediate. Conversely, it is possible to show that any w that satisfies *gnd*$(\phi)|H_1^+ \wedge \neg\alpha$ can be extended to a world that satisfies *gnd*$(\phi) \wedge \neg\alpha$ which agrees with w on $\{f(\boldsymbol{m}) \mid m_j \in H_1^+\}$. Now, by applying Theorem 8 and then Theorem 9, we are able to obtain:

Corollary 1: *Let ϕ be a proper$^+$KB (restricted as above), α be a QF fluent formula with a single free variable x and H is as above. Then $\phi \models \forall\alpha$ iff gnd$(\phi)|C_1^+ \models \alpha_m^x$ for all $m \in C$, where $C = H_1^+$.*

Example 2: **(a.)** If α and β denote $\rho \neq K$ and $\rho = Q$ resp., $\alpha \not\models \neg\beta$ since $H(\alpha \wedge \beta, \rho) = \{K, Q\}$, $[\alpha \wedge \beta]_{\rho=Q} =$ SAT. And since $[\alpha \wedge \neg\beta]_{\rho=n} =$ SAT for any new name n, $\alpha \not\models \beta$.

(**b.**) Consider ϕ_1 from Ex. 1 and let $\alpha = \{in(c_2) = s_1\}$. To see how $\phi_1 \not\models \alpha$, let $H_1^+ = \{5, s_1, c_1, c_2, J, Q, K, n\}$ where n is new. Now consider $\delta =$ *gnd*$(\phi_1)|H_1^+ \wedge \neg\alpha$. The only equality mentioning *in*(c_2) in δ is $\neg\alpha$. Since *gnd*$(\phi_1)|H_1^+$ is *sat.*, let *in*$(c_2) = m$, $m \in \mathcal{N} - \{s_1\}$ and we obtain SAT.

(**c.**) Let α be *in*$(x) = s_1 \supset$ *value*$(x) \neq 4$. Here, $\phi_1 \models \forall\alpha$. To see this let C be H_1^+ from above. Then we obtain UNSAT with *gnd*$(\phi_1)|C_1^+ \wedge \neg(\alpha_m^x)$ for all substitutions $m \in C$.

Conclusions

We have presented the following results. We consider the expressive proper$^+$KBs, but this time extended to include functions.[16] With proper$^+$KBs as initial theories, we show FO definability results for local-effect, normal and range-restricted theories. These cover all syntactically restricted action theories studied so far. Second, we prove efficient computability results for all three cases. Third, we consider the query evaluation problem, and show that despite the expressiveness, a sound and complete decision procedure exists for a large class of queries. The procedure shares structural properties with satisfiability solvers, and yet quantification is interpreted over an infinite universe. Given the success of Boolean reasoners, we believe these results show promise especially as far as progression is concerned in expressive settings. We remark that besides the LR definition, a number of alternate ones appears in the literature [Lakemeyer and Levesque, 2009].

While the chosen formalism in the paper is the situation calculus, we do believe our knowledge bases can be used with others. We think an important direction for future work lies in identifying classes of action theories for which the *regressed* formula [Reiter, 2001] remains in a form that is decidable.

As a concluding remark, reasoning about functions, equalities and constants is certainly not a new effort, and goes back to Ackermann [Badban *et al.*, 2007]. Here too one finds variants of DPLL for certain QF fragments of FO logic, where often nesting of functions is allowed and the uniqueness of constants is not assumed. However, it is the latter property that allows us to handle quantifiers, both in the query and the KB, in a decidable manner. We believe the formalism provides a new insight in terms of finding satisfying assignments to ground functions. Nonetheless, it remains to be seen if it is the more practically viable option, and should be favored to say an encoding of the KB (plus, the uniqueness of names) input to the many solvers [Badban *et al.*, 2007].

[15] To see why such a restriction is needed, let $\phi = \forall.x \neq n \supset f(n) \neq x$ for some n. While ϕ entails $f(n) = n$, *gnd*$(\phi)|H_1^+$ does not. If we add, say, $f(n) = n \vee f(n) = m$ to ϕ then *gnd*$(\phi)|H_1^+$ does entail $f(n) = n$ as required. However, if there are only finitely many inequalities, then they will also appear in *gnd*$(\phi)|H_1^+$.

[16] Proper$^+$KBs with functions are strictly more expressive. For instance, although formulas of the form $\forall.x \neq y \supset \neg P(x) \vee \neg P(y)$ simulate the *uniqueness* of function values, we can not capture the *existence* of values, *i.e.* $\exists x.P(x)$, due to the infinite discourse.

Acknowledgements

We thank anonymous reviewers for helpful suggestions. The first author is supported by the B-IT Graduate School.

References

[Badban *et al.*, 2007] B. Badban, J. v. de Pol, O. Tveretina, and H. Zantema. Generalizing DPLL and satisfiability for equalities. *Inf. Comput.*, 205:1188–1211, 2007.

[Belle, 2011] V. Belle. Ph.D. Dissertation, Dept. of Computer Science, RWTH Aachen University, 2011. In Preparation.

[Davis and Putnam, 1960] M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 1960.

[Doherty *et al.*, 2001] P. Doherty, W. Lukaszewicz, and A. Szalas. Computing Strongest Necessary and Weakest Sufficient Conditions of First-Order Formulas. In *Proc. IJCAI-01*, 2001.

[Gomes *et al.*, 2008] C. Gomes, H. Kautz, A. Sabharwal, and B. Selman. Satisfiability solvers. In *Handbook of KR*. Elsevier, 2008.

[Hughes and Cresswell, 1972] G. E. Hughes and M. J. Cresswell. *An introduction to modal logic*. Methuen London, 1972.

[Kautz and Selman, 1992] H. Kautz and B. Selman. Planning as satisfiability. In *Proc. ECAI-92*, 1992.

[Lakemeyer and Levesque, 2002] G. Lakemeyer and H. Levesque. Evaluation-based reasoning with disjunctive information in first-order knowledge bases. In *Proc. KR*, 2002.

[Lakemeyer and Levesque, 2009] G. Lakemeyer and H. Levesque. A semantical account of progression in the presence of defaults. *Conceptual Modeling: Foundations and Applications*, 2009.

[Levesque and Lakemeyer, 2001] H. Levesque and G. Lakemeyer. *The logic of knowledge bases*. The MIT Press, 2001.

[Lin and Reiter, 1994] F. Lin and R. Reiter. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, 1994.

[Lin and Reiter, 1997] F. Lin and R. Reiter. How to progress a database. *Artificial Intelligence*, 92(1-2):131–167, 1997.

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

[Liu *et al.*, 2004] Y. Liu, G. Lakemeyer, and H. Levesque. A logic of limited belief for reasoning with disjunctive information. In *Proc. KR*, 2004.

[Reiter, 2001] R. Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.

[Vassos *et al.*, 2009] S. Vassos, S. Sardina, and H. Levesque. Progressing basic action theories with non-local effect actions. In *Proc. Commonsense Reasoning*, 2009.