On First-Order Definability and Computability of Progression for Local-Effect Actions and Beyond

Yongmei Liu
Dept. of Computer Science
Sun Yat-sen University
Guangzhou 510275, China
ymliu@mail.sysu.edu.cn

Gerhard Lakemeyer
Dept. of Computer Science
RWTH Aachen
52056 Aachen, Germany
gerhard@cs.rwth-aachen.de

Abstract

In a seminal paper, Lin and Reiter introduced the notion of progression for basic action theories in the situation calculus. Unfortunately, progression is not first-order definable in general. Recently, Vassos, Lakemeyer, and Levesque showed that in case actions have only local effects, progression is first-order representable. However, they could show computability of the first-order representation only for a restricted class. Also, their proofs were quite involved. In this paper, we present a result stronger than theirs that for local-effect actions, progression is always first-order definable and computable. We give a very simple proof for this via the concept of forgetting. We also show first-order definability and computability results for a class of knowledge bases and actions with non-local effects. Moreover, for a certain class of local-effect actions and knowledge bases for representing disjunctive information, we show that progression is not only first-order definable but also efficiently computable.

1 Introduction

A fundamental problem in reasoning about action is *projection*, which is concerned with determining whether or not a formula holds after a number of actions have occurred, given a description of the preconditions and effects of the actions and what the world is like initially. Projection plays an important role in planning and in action languages such as Golog [Levesque *et al.*, 1997] or $\mathcal{A}$ [Gelfond and Lifschitz, 1993].

Two powerful methods to solve the projection problem are *regression* and *progression*. Roughly, regression reduces a query about the future to a query about the initial knowledge base (KB). Progression, on the other hand, changes the initial KB according to the effects of each action and then checks whether the formula holds in the resulting KB. One advantage of progression compared to regression is that after a KB has been progressed, many queries about the resulting state can be processed without any extra overhead. Moreover, when the action sequence becomes very long, as in the case of a robot operating for an extended period of time, regression simply becomes unmanageable. However, projection via progression has three main computational requirements which are not easy to satisfy: the new KB must be efficiently computed, its size should be at most linear in the size of the initial KB (to allow for iterated progression), and it must be possible to answer the query efficiently from the new KB.

As Lin and Reiter [1997] showed in the framework of Reiter's version of the situation calculus [Reiter, 2001], progression is second order in general. And even if it is first-order (FO) definable, the size of the progressed KB may be unmanageable and even infinite. Recently, Vassos and Levesque [2008] also showed that the second-order nature of progression is in general inescapable, as a restriction to FO theories (even infinite ones) is strictly weaker in the sense that inferences about the future may be lost compared to the second-order version.

Nevertheless, for restricted action theories, progression can be FO definable and very effective. The classical example is STRIPS, where the initial KB is a set of literals and progression can be described via the usual *add* and *delete* lists. Since STRIPS is quite limited in expressiveness, it seems worthwhile to investigate more powerful action descriptions which still lend themselves to FO definable progression. Lin and Reiter [1997] already identified two such cases, and recently Vassos *et al.* [2008] were able to show that for so-called local-effect actions, which only change the truth values of fluent atoms with arguments mentioned by the actions, progression is always FO definable. However, they showed the computability of the FO representation only for a special case.

In this paper, we substantially improve and extend the results of Vassos *et al.*. By appealing to the notion of forgetting [Lin and Reiter, 1994], we show that progression for arbitrary local-effect actions is always FO definable and computable. We extend this result to certain actions with non-local effects like the briefcase domain, where moving a briefcase implicitly moves all the objects contained in it. For the special case of so-called proper$^+$ KBs [Lakemeyer and Levesque, 2002] and a restricted class of local-effect actions we show that progression is not only first-order definable but also efficiently computable.

The rest of the paper is organized as follows. In the next section we introduce background material, including the notion of forgetting, Reiter's basic action theories, and progression. In Section 3 we present our result concerning local-effect actions. Section 4 deals with non-local effects and Section 5 with proper$^+$ KBs. Then we conclude.

Dagstuhl Seminar Proceedings 10081
Cognitive Robotics
http://drops.dagstuhl.de/opus/volltexte/2010/2638

2 Preliminaries

We start with a first-order language $\mathcal{L}$ with equality. The set of formulas of $\mathcal{L}$ is the least set which contains the atomic formulas, and if ϕ and ψ are in the set and x is a variable, then $\neg\phi$, $(\phi \wedge \psi)$ and $\forall x\phi$ are in the set. The connectives $\vee$, $\supset$, $\equiv$, and $\exists$ are understood as the usual abbreviations. To improve readability, sometimes we put parentheses around quantifiers. We use the "dot" notation to indicate that the quantifier preceding the dot has maximum scope, *e.g.*, $\forall x.P(x) \supset Q(x)$ stands for $\forall x[P(x) \supset Q(x)]$. We often omit leading universal quantifiers in writing sentences. We use $\phi \Leftrightarrow \psi$ to mean that ϕ and ψ are logically equivalent. Let ϕ be a formula, and let μ and μ' be two expressions. We denote by $\phi(\mu/\mu')$ the result of replacing every occurrence of μ in ϕ with μ'.

2.1 Forgetting

Lin and Reiter [1994] defined the concept of forgetting a ground atom or predicate in a theory. Intuitively, the resulting theory should be weaker than the original one, but entail the same set of sentences that are "irrelevant" to the ground atom or predicate.

Definition 2.1 Let μ be either a ground atom $P(\vec{t})$ or a predicate symbol P. Let M_1 and M_2 be two structures. We write $M_1 \sim_\mu M_2$ if M_1 and M_2 agree on everything except possibly on the interpretation of μ.

Definition 2.2 Let T be a theory, and μ a ground atom or predicate. A theory T' is a result of forgetting μ in T, denoted by forget$(T, \mu) \Leftrightarrow T'$, if for any structure M, $M \models T'$ iff there is a model M' of T such that $M \sim_\mu M'$.

Clearly, if both T' and T'' are results of forgetting μ in T, then they are logically equivalent. Similarly, we can define the concept of forgetting a set of atoms or predicates. In this paper, we are only concerned with finite theories. So henceforth we only deal with forgetting for sentences.

Lin and Reiter [1994] showed that for any sentence ϕ and atom p, forgetting p in ϕ is FO definable and can be obtained from ϕ and p by simple syntactic manipulations. Here we reformulate their result in the context of forgetting a finite number of atoms. We first introduce some notation.

Let Γ be a finite set of ground atoms. We call a truth assignment θ of atoms from Γ a Γ-model. Clearly, a Γ-model θ can be represented by a conjunction of literals. We use $\mathcal{M}(\Gamma)$ to denote the set of all Γ-models. Let ϕ be a formula, and θ a Γ-model. We use $\phi[\theta]$ to denote the result of replacing every occurrence of $P(\vec{t})$ in ϕ by the following formula:

$$\bigvee_{j=1}^{m} (\vec{t} = \vec{t}_j \wedge v_j) \vee (\bigwedge_{j=1}^{m} \vec{t} \neq \vec{t}_j) \wedge P(\vec{t}),$$

where for $j = 1, \ldots, m$, the truth value of $P(\vec{t}_j)$ is specified by θ, and v_j is the truth value.

Proposition 2.3 *Let ϕ be a formula, $M \sim_\Gamma M'$, $\theta \in \mathcal{M}(\Gamma)$, and $M \models \theta$. Then for any variable assignment σ, $M, \sigma \models \phi$ iff $M', \sigma \models \phi[\theta]$.*

Theorem 2.4 *forget$(\phi, \Gamma) \Leftrightarrow \bigvee_{\theta \in \mathcal{M}(\Gamma)} \phi[\theta]$.*

Proof: Let M be a structure. We show that $M \models \bigvee_{\theta \in \mathcal{M}(\Gamma)} \phi[\theta]$ iff there is a model M' of ϕ s.t. $M \sim_\Gamma M'$. Suppose the latter. Let $\theta \in \mathcal{M}(\Gamma)$ s.t. $M' \models \theta$. By Proposition 2.3, $M \models \phi[\theta]$. Now suppose $M \models \phi[\theta]$, where $\theta \in \mathcal{M}(\Gamma)$. Let M' be the structure s.t. $M \sim_\Gamma M'$ and $M' \models \theta$. By Proposition 2.3, $M' \models \phi$. ■

Corollary 2.5
forget$(\phi, \Gamma) \Leftrightarrow \bigvee_{\theta \in \mathcal{M}(\Gamma)}$ and $\phi \wedge \theta$ is consistent $\phi[\theta]$.

Proof: By Proposition 2.3, $\phi \wedge \theta$ entails $\phi[\theta]$.
Thus if $\phi \wedge \theta$ is inconsistent, so is $\phi[\theta]$. ■

Example 2.1 Let $\phi = \forall x.clear(x)$,
and $\Gamma = \{clear(A), clear(B)\}$. Then
$\phi[clear(A) \wedge clear(B)] = \forall x.x = A \wedge true \vee x = B \wedge true$
$\vee\ x \neq A \wedge x \neq B \wedge clear(x)$, which is equivalent to
$\forall x.x = A \vee x = B \vee x \neq A \wedge x \neq B \wedge clear(x)$. Similarly,
$\phi[clear(A) \wedge \neg clear(B)] \Leftrightarrow$
$\forall x.x = A \vee x \neq A \wedge x \neq B \wedge clear(x)$,
$\phi[\neg clear(A) \wedge clear(B)] \Leftrightarrow$
$\forall x.x = B \vee x \neq A \wedge x \neq B \wedge clear(x)$, and
$\phi[\neg clear(A) \wedge \neg clear(B)] \Leftrightarrow$
$\forall x.x \neq A \wedge x \neq B \wedge clear(x)$.
Thus forget$(\phi, \Gamma) \Leftrightarrow$
$\forall x.x = A \vee x = B \vee x \neq A \wedge x \neq B \wedge clear(x)$,
which is equivalent to $\forall x.x \neq A \wedge x \neq B \supset clear(x)$.

We now assume $\mathcal{L}^2$, the second-order extension of $\mathcal{L}$.

Theorem 2.6 *forget$(\phi, P) \Leftrightarrow \exists R.\phi(P/R)$, where R is a second-order predicate variable.*

Example 2.2 Let $\phi_1 = \forall x.clear(x) \vee \exists y.on(y, x)$. Then forget$(\phi_1, clear) \Leftrightarrow \exists R \forall x.R(x) \vee \exists y.on(y, x)$, which is equivalent to $true$. Let $\phi_2 = \exists x.clear(x) \wedge \exists y.on(x, y)$. Then forget$(\phi_2, clear) \Leftrightarrow \exists R \exists x.R(x) \wedge \exists y.on(x, y)$, which is equivalent to $\exists x \exists y.on(x, y)$.

In general, forgetting a predicate is not FO definable. Naturally, by Theorem 2.6, second-order quantifier elimination techniques can be used for obtaining FO definability results for forgetting a predicate. In fact, Doherty *et al.* [2001] used such techniques for computing strongest necessary and weakest sufficient conditions of FO formulas, which are concepts closely related to forgetting. As surveyed in [Nonnengart *et al.*, 1999], the following is a classical result on second-order quantifier elimination due to Ackermann (1935). A formula ϕ is positive (resp. negative) wrt a predicate P if $\neg P$ (resp. P) does not occur in the negation normal form of ϕ.

Theorem 2.7 *Let P be a predicate variable, and let ϕ and $\psi(P)$ be FO formulas such that $\psi(P)$ is positive wrt P and ϕ contains no occurrence of P at all. Then*

$$\exists P. \forall \vec{x}(\neg P(\vec{x}) \vee \phi(\vec{x})) \wedge \psi(P)$$

is logically equivalent to $\psi(P(\vec{x}) \leftarrow \phi(\vec{x}))$, denoting the result of replacing each occurrence of $P(\vec{t})$ in ψ with $\phi(\vec{t})$, and similarly if the sign of P is switched and ψ is negative wrt P.

2.2 Basic action theories

The language $\mathcal{L}_{sc}$ of the situation calculus [Reiter, 2001] is a many-sorted first-order language suitable for describing dynamic worlds. There are three disjoint sorts: $action$ for actions, $situation$ for situations, and $object$ for everything else. $\mathcal{L}_{sc}$ has the following components: a constant S_0 denoting the initial situation; a binary function $do(a, s)$ denoting the successor situation to s resulting from performing action a; a binary predicate $Poss(a, s)$ meaning that action a is possible in situation s; action functions, *e.g.*, $move(x, y)$; a finite number of relational fluents, *i.e.*, predicates taking a situation term as their last argument, *e.g.*, $ontable(x, s)$; and a finite number of situation-independent predicates and functions. For simplicity of presentation, we do not consider functional fluents in this paper.

Often, we need to restrict our attention to formulas that refer to a particular situation. For this purpose, we say that a formula ϕ is uniform in a situation term τ, if ϕ does not mention any other situation terms except τ, does not quantify over situation variables, and does not mention $Poss$.

A particular domain of application will be specified by a basic action theory of the following form:

$$\mathcal{D} = \Sigma \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0}, \text{ where}$$

1. Σ is the set of the foundational axioms for situations.
2. $\mathcal{D}_{ap}$ is a set of action precondition axioms.
3. $\mathcal{D}_{ss}$ is a set of successor state axioms (SSAs), one for each fluent, of the form

   $$F(\vec{x}, do(a, s)) \equiv \gamma^+_F(\vec{x}, a, s) \vee (F(\vec{x}, s) \wedge \neg\gamma^-_F(\vec{x}, a, s)),$$

   where $\gamma^+_F(\vec{x}, a, s)$ and $\gamma^-_F(\vec{x}, a, s)$ are uniform in s.
4. $\mathcal{D}_{una}$ is the set of unique names axioms for actions: $A(\vec{x}) \neq A'(\vec{y})$, and $A(\vec{x}) = A(\vec{y}) \supset \vec{x} = \vec{y}$, where A and A' are distinct action functions.
5. $\mathcal{D}_{S_0}$, usually called the initial database, is a finite set of sentences uniform in S_0. We call $\mathcal{D}_{S_0}$ the initial KB.

2.3 Progression

Lin and Reiter [1997] formalized the notion of progression. Let $\mathcal{D}$ be a basic action theory, and α a ground action. We denote by S_α the situation term $do(\alpha, S_0)$.

Definition 2.8 Let M and M' be structures with the same domains for sorts *action* and *object*. We write $M \sim_{S_\alpha} M'$ if the following two conditions hold: (1) M and M' interpret all situation-independent predicate and function symbols identically. (2) M and M' agree on all fluents at S_α: For every relational fluent F, and every variable assignment σ, $M, \sigma \models F(\vec{x}, S_\alpha)$ iff $M', \sigma \models F(\vec{x}, S_\alpha)$.

We denote by $\mathcal{L}^2_{sc}$ the second-order extension of $\mathcal{L}_{sc}$. The notion of uniform formulas carries over to $\mathcal{L}^2_{sc}$.

Definition 2.9 Let $\mathcal{D}_{S_\alpha}$ be a set of sentences in $\mathcal{L}^2_{sc}$ uniform in S_α. $\mathcal{D}_{S_\alpha}$ is a progression of the initial KB $\mathcal{D}_{S_0}$ wrt α if for any structure M, M is a model of $\mathcal{D}_{S_\alpha}$ iff there is a model M' of $\mathcal{D}$ such that $M \sim_{S_\alpha} M'$.

Lin and Reiter [1997] proved that progression is always second-order definable. They used an old version of SSAs in their formulation of the result; here we reformulate their result using the current form of SSAs.

We let $\mathcal{D}_{ss}[\alpha, S_0]$ denote the instantiation of $\mathcal{D}_{ss}$ wrt α and S_0, *i.e.*, the set of sentences $F(\vec{x}, do(\alpha, S_0)) \equiv \Phi_F(\vec{x}, \alpha, S_0)$. Let $F_1, \ldots, F_n$ be all the fluents. We introduce n new predicate symbols $P_1, \ldots, P_n$. We use $\phi \uparrow S_0$ to denote the result of replacing every occurrence of $F_i(\vec{t}, S_0)$ in ϕ by $P_i(\vec{t})$. We call P_i the lifting predicate for F_i. When Σ is a finite set of formulas, we denote by $\wedge\Sigma$ the conjunction of its elements.

Theorem 2.10 *The following is a progression of $\mathcal{D}_{S_0}$ wrt α:*

$$\exists\vec{R}.\{\bigwedge(\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{ss}[\alpha, S_0]) \uparrow S_0\}(\vec{P}/\vec{R}),$$

where R_1, ..., R_n are second-order predicate variables.

Therefore, by Theorem 2.6, if ϕ is uniform in S_α and ϕ is a result of forgetting the lifting predicates $\vec{P}$ in $\bigwedge(\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{ss}[\alpha, S_0]) \uparrow S_0$, then it is a progression of $\mathcal{D}_{S_0}$ wrt α.

3 Progression for local-effect actions

In this section, we show that for local-effect actions, progression is always FO-definable and computable.

We first show an intuitive result concerning forgetting a predicate: if a sentence ϕ entails that the truth values of two predicates P and Q are different at only a finite number of certain instances, then forgetting the predicate Q in ϕ can be obtained from forgetting the Q atoms of these instances in ϕ and then replacing Q by P in the result.

Let $\vec{x}$ be a variable vector, and let $\Delta = \{\vec{t}_1, \ldots, \vec{t}_m\}$ be a set of vectors of ground terms, where all the vectors have the same length. We use $\vec{x} \in \Delta$ to denote the formula $\vec{x} = \vec{t}_1 \vee \ldots \vee \vec{x} = \vec{t}_m$. Let P and Q be two predicates. We let $Q(\Delta)$ denote the set $\{Q(\vec{t}) \mid \vec{t} \in \Delta\}$, and we let $P \approx_\Delta Q$ denote the sentence $\forall\vec{x}.\vec{x} \notin \Delta \supset P(\vec{x}) \equiv Q(\vec{x})$.

Proposition 3.1 *Let ϕ be a formula, $M \models P \approx_\Delta Q$, and $\theta \in \mathcal{M}(Q(\Delta))$. Then for any variable assignment σ, $M, \sigma \models \phi[\theta](Q/P)$ iff $M, \sigma \models \phi[\theta]$.*

Theorem 3.2 *Let P and Q be two predicates, and Δ a finite set of vectors of ground terms. If forget$(\phi, Q(\Delta)) \Leftrightarrow \psi$, then forget$(\phi \wedge (P \approx_\Delta Q), Q) \Leftrightarrow \psi(Q/P)$.*

Proof: Let $\Gamma = Q(\Delta)$. By Theorem 2.4, forget$(\phi, \Gamma) \Leftrightarrow \bigvee_{\theta \in \mathcal{M}(\Gamma)} \phi[\theta]$. Let M be a structure. We show that $M \models \bigvee_{\theta \in \mathcal{M}(\Gamma)} \phi[\theta](Q/P)$ iff there is a model M' of $\phi \wedge (P \approx_\Delta Q)$ s.t. $M \sim_Q M'$. Suppose the latter. Let $\theta \in \mathcal{M}(\Gamma)$ s.t. $M' \models \theta$. Since $M' \models \theta$ and $M' \models \phi$, by Proposition 2.3, $M' \models \phi[\theta]$. Since $M' \models P \approx_\Delta Q$, by Proposition 3.1, $M' \models \phi[\theta](Q/P)$. Since $M \sim_Q M'$, $M \models \phi[\theta](Q/P)$.

Now suppose $M \models \phi[\theta](Q/P)$, where $\theta \in \mathcal{M}(\Gamma)$. Let M' be the structure s.t. $M \sim_Q M'$, $M' \models P \approx_\Delta Q$, and $M' \models \theta$. Since $M \sim_Q M'$ and $M \models \phi[\theta](Q/P)$, $M' \models \phi[\theta](Q/P)$. Since $M' \models P \approx_\Delta Q$, by Proposition 3.1, $M' \models \phi[\theta]$. Since $M' \models \theta$, by Proposition 2.3, $M' \models \phi$. ■

Actions in many dynamic domains have only local effects in the sense that if an action $A(\vec{c})$ changes the truth value

of an atom $F(\vec{d}, s)$, then $\vec{d}$ is contained in $\vec{c}$. This contrasts with actions having non-local effects such as moving a briefcase, which will also move all the objects inside the briefcase without having mentioned them.

Definition 3.3 An SSA is local-effect if both $\gamma_F^+(\vec{x}, a, s)$ and $\gamma_F^-(\vec{x}, a, s)$ are disjunctions of formulas of the form $\exists \vec{z}[a = A(\vec{u}) \wedge \phi(\vec{u}, s)]$, where A is an action function, $\vec{u}$ contains $\vec{x}$, $\vec{z}$ is the remaining variables of $\vec{u}$, and ϕ is called a context formula. An action theory is local-effect if each SSA is local-effect.

Example 3.1 Consider a simple blocks world. We use a single action, $move(x, y, z)$, moving a block x from block y to block z. We use two fluents: $clear(x, s)$, block x has no blocks on top of it; $on(x, y, s)$, block x is on block y; $eh(x, s)$, the height of block x is even. Clearly, the following SSAs are local-effect:

$clear(x, do(a, s)) \equiv (\exists y, z)a = move(y, x, z) \vee$
$\quad clear(x, s) \wedge \neg(\exists y, z)a = move(y, z, x),$
$on(x, y, do(a, s)) \equiv (\exists z)a = move(x, z, y) \vee$
$\quad on(x, y, s) \wedge \neg(\exists z)a = move(x, y, z),$
$eh(x, do(a, s)) \equiv (\exists y, z)[a = move(x, y, z) \wedge \neg eh(z, s)] \vee$
$\quad eh(x, s) \wedge \neg(\exists y, z)[a = move(x, y, z) \wedge eh(z, s)].$

By using the unique names axioms, the instantiation of a local-effect SSA on a ground action can be simplified. Suppose the SSA for F is local-effect. Let $\alpha = A(\vec{t})$ be a ground action. Then each of $\gamma_F^+(\vec{x}, \alpha, s)$ and $\gamma_F^-(\vec{x}, \alpha, s)$ is equivalent to a formula of the following form:

$$\vec{x} = \vec{t}_1 \wedge \psi_1(s) \vee \ldots \vee \vec{x} = \vec{t}_n \wedge \psi_n(s),$$

where $\vec{t}_i$ is a vector of ground terms contained in $\vec{t}$, and $\psi_i(s)$ is a formula whose only free variable is s. Without loss of generality, we assume that for a local-effect SSA, $\gamma_F^+(\vec{x}, \alpha, s)$ and $\gamma_F^-(\vec{x}, \alpha, s)$ have the above simplified form. In the case of our blocks world, we have:

$clear(x, do(move(c_1, c_2, c_3), s)) \equiv x = c_2 \vee$
$\quad clear(x, s) \wedge \neg(x = c_3),$
$on(x, y, do(move(c_1, c_2, c_3), s)) \equiv x = c_1 \wedge y = c_3 \vee$
$\quad on(x, y, s) \wedge \neg(x = c_1 \wedge y = c_2),$
$eh(x, do(move(c_1, c_2, c_3), s)) \equiv x = c_1 \wedge \neg eh(c_3, s) \vee$
$\quad eh(x, s) \wedge \neg(x = c_1 \wedge eh(c_3, s)).$

Following Vassos *et al.* [2008], we define the concepts of argument set and characteristic set:

Definition 3.4 Let $\mathcal{D}$ be local-effect, and α a ground action. The argument set of fluent F wrt α is the following set:

$$\Delta_F = \{\vec{t} \mid \vec{x} = \vec{t} \text{ appears in } \gamma_F^+(\vec{x}, \alpha, s) \text{ or } \gamma_F^-(\vec{x}, \alpha, s)\}.$$

The characteristic set of α is the following set of atoms:

$$\Omega(s) = \{F(\vec{t}, s) \mid F \text{ is a fluent and } \vec{t} \in \Delta_F\}.$$

We let $\mathcal{D}_{ss}[\Omega]$ denote the instantiation of $\mathcal{D}_{ss}$ wrt Ω, *i.e.*, the set of sentences $F(\vec{t}, S_\alpha) \equiv \Phi_F(\vec{t}, \alpha, S_0)$, where $F(\vec{t}, s) \in \Omega$. We let $\mathcal{D}_{ss}[\overline{\Omega}]$ denote the set of sentences $\vec{x} \notin \Delta_F \supset F(\vec{x}, S_\alpha) \equiv F(\vec{x}, S_0)$. Then we have

Proposition 3.5 *Let $\mathcal{D}$ be local-effect, and α a ground action. Then $\mathcal{D}_{una} \models \mathcal{D}_{ss}[\alpha, S_0] \equiv \mathcal{D}_{ss}[\Omega] \cup \mathcal{D}_{ss}[\overline{\Omega}]$.*

Theorem 3.6 *Let $\mathcal{D}$ be local-effect, and α a ground action. Let $\Omega(s)$ be the characteristic set of α. Then the following is a progression of $\mathcal{D}_{S_0}$ wrt α:*

$$\bigwedge \mathcal{D}_{una} \wedge \bigvee_{\theta \in \mathcal{M}(\Omega(S_0))} (\mathcal{D}_{S_0} \cup \mathcal{D}_{ss}[\Omega])[\theta](S_0/S_\alpha).$$

Proof: By Proposition 3.5, Theorems 2.4, 2.10 and 3.2. ■

The size of the progression is $O(2^m n)$, where m is the size of the characteristic set, and n is the size of the action theory. When we do iterative progression wrt a sequence δ of actions, the size of the resulting KB is $O(2^{lm} n)$, where l is the length of δ, and m is the maximum size of the characteristic sets.

Corollary 3.7 *Progression for local-effect actions is always FO definable and computable.*

We remark that this result is strictly more general than the one obtained by Vassos *et al.* [2008], who only showed that progression for local-effect actions is FO definable in a non-constructive way, *i.e.*, they left open the question whether the FO representation is computable or even finite. Moreover, while their proof is quite involved, having to appeal to Compactness of FO logic, ours is actually fairly simple.

Example 3.1 continued. Let $\alpha = move(A, B, C)$. Then $\Omega = \{clear(B, s), clear(C, s), on(A, B, s), on(A, C, s), eh(A, s)\}$. $\mathcal{D}_{ss}[\Omega]$ is simplified to $\{clear(B, S_\alpha), \neg clear(C, S_\alpha), \neg on(A,B,S_\alpha), on(A,C,S_\alpha), eh(A,S_\alpha) \equiv \neg eh(C, S_0)\}$.

Let $\mathcal{D}_{S_0}$ be the following set of sentences:
$A \neq B, A \neq C, B \neq C, clear(A, S_0),$
$on(A, B, S_0), clear(C, S_0), clear(x, S_0) \supset eh(x, S_0),$
$on(x, y, S_0) \supset \neg clear(y, S_0).$
Then $\mathcal{D}_{S_0}$ entails ϑ, denoting $\neg clear(B, S_0) \wedge clear(C, S_0) \wedge on(A, B, S_0) \wedge \neg on(A, C, S_0)$. Thus there are only two $\theta \in \mathcal{M}(\Omega(S_0))$ which are consistent with $\mathcal{D}_{S_0}$:
$\theta_1 = \vartheta \wedge eh(A, S_0)$ and $\theta_2 = \vartheta \wedge \neg eh(A, S_0)$.

For example, let $\phi = clear(x, S_0) \supset eh(x, S_0)$.
Then $\phi[\theta_1] \Leftrightarrow clear(x, S_0)[\vartheta] \supset x = A \vee x \neq A \wedge eh(x, S_0)$,
and $\phi[\theta_2] \Leftrightarrow clear(x, S_0)[\vartheta] \supset x \neq A \wedge eh(x, S_0)$.
Thus $\phi[\theta_1] \vee \phi[\theta_2]$ is equivalent to
$x = C \vee x \neq B \wedge x \neq C \wedge clear(x, S_0)$
$\quad \supset x = A \vee x \neq A \wedge eh(x, S_0).$

By Theorem 3.6 and Corollary 2.5, the following is a progression of $\mathcal{D}_{S_0}$ wrt α:
$move(x_1, y_1, z_1) = move(x_2, y_2, z_2)$
$\quad \supset x_1 = x_2 \wedge y_1 = y_2 \wedge z_1 = z_2,$
$A \neq B, A \neq C, B \neq C, clear(A, S_\alpha),$
$x = C \vee x \neq B \wedge x \neq C \wedge clear(x, S_\alpha) \supset$
$\quad x = A \vee x \neq A \wedge eh(x, S_\alpha),$
$x = A \wedge y = B \vee$
$\quad (x \neq A \vee y \neq B) \wedge (x \neq A \vee y \neq C) \wedge on(x, y, S_\alpha) \supset$
$\quad \neg(y = C \vee y \neq B \wedge y \neq C \wedge clear(y, S_\alpha)),$
$clear(B, S_\alpha), \neg clear(C, S_\alpha), \neg on(A, B, S_\alpha),$
$on(A, C, S_\alpha), eh(A, S_\alpha) \equiv \neg eh(C, S_\alpha).$

4 Progression for normal actions

In the last section, we showed that for local-effect actions, progression is FO definable and computable. An interesting observation about non-local-effect actions is that their effects often do not depend on the fluents on which they have non-local effects, that is, they normally have local effects on the

fluents that appear in every γ_F^+ and γ_F^-. For example, moving a briefcase will move all the objects in it as well without affecting the fluent *in*. We will call such an action a normal action. In this section, we show that for a normal action α, if the initial KB has the property that for each fluent F on which α has non-local effects, the only appearance of F is in the form of $\phi(\vec{x}) \supset F(\vec{x}, S_0)$ or $\phi(\vec{x}) \supset \neg F(\vec{x}, S_0)$, then progression is FO definable and computable.

Our result is inspired by a result by Lin and Reiter [1997] that for context-free action theories, that is, action theories where every predicate appearing in every γ_F^+ and γ_F^- is situation-independent, if the initial KB has the property that for each fluent F, the only appearance of F is in the form of $\phi(\vec{x}) \supset F(\vec{x}, S_0)$ or $\phi(\vec{x}) \supset \neg F(\vec{x}, S_0)$, then progression is FO definable and computable. Incidentally, their result can be considered as an application of a simple case of the classical result by Ackermann (see Theorem 2.7). To prove our result, we combine the application of this simple case and the proof idea behind our result for local-effect actions.

We first present this simple case of Ackermann's result:

Definition 4.1 We say that a finite theory T is semi-definitional wrt a predicate P if the only appearance of P in T is in the form of $P(\vec{x}) \supset \phi(\vec{x})$, where we call $\phi(\vec{x})$ a necessary condition of P, or $\phi(\vec{x}) \supset P(\vec{x})$, where we call $\phi(\vec{x})$ a sufficient condition of P. We use WSC_P (meaning weakest sufficient condition) to denote the disjunction of $\phi(\vec{x})$ such that $\phi(\vec{x}) \supset P(\vec{x})$ is in T, and we use SNC_P (meaning strongest necessary condition) to denote the conjunction of $\phi(\vec{x})$ such that $P(\vec{x}) \supset \phi(\vec{x})$ is in T.

Theorem 4.2 *Let T be finite and semi-definitional wrt P. Let T' be the set of sentences in T that contains no occurrence of P. Then forget$(T, P) \Leftrightarrow T' \wedge \forall \vec{x}.\text{WSC}_P(\vec{x}) \supset \text{SNC}_P(\vec{x})$.*

Proof: Clearly, $\exists R.T(P/R) \models T' \wedge \forall \vec{x}.\text{WSC}_P(\vec{x}) \supset \text{SNC}_P(\vec{x})$. To prove the opposite entailment, simply use the definition $\forall \vec{x}.P(\vec{x}) \equiv \text{WSC}_P(\vec{x})$. ■

The following proposition shows that the SSA for a fluent F is semi-definitional wrt the predicate $F(\vec{x}, S_0)$ provided that F does not appear in γ_F^+ or γ_F^-.

Proposition 4.3 *The sentence $F(\vec{x}, S_\alpha) \equiv \gamma_F^+(\vec{x}, \alpha, S_0) \vee F(\vec{x}, S_0) \wedge \neg\gamma_F^-(\vec{x}, \alpha, S_0)$ is equivalent to the following sentences: $\neg\gamma_F^+ \wedge F(\vec{x}, S_\alpha) \supset F(\vec{x}, S_0)$, $F(\vec{x}, S_0) \supset \gamma_F^- \vee F(\vec{x}, S_\alpha)$, $\gamma_F^+ \supset F(\vec{x}, S_\alpha)$, and $\neg\gamma_F^+ \wedge \gamma_F^- \supset \neg F(\vec{x}, S_\alpha)$.*

We now formalize our constraints on the actions and the initial KBs.

Definition 4.4 We say that a ground action α has local effects on a fluent F, if by using $\mathcal{D}_{una}$, each of $\gamma_F^+(\vec{x}, \alpha, s)$ and $\gamma_F^-(\vec{x}, \alpha, s)$ can be simplified to a disjunction of formulas of the form $\vec{x} = \vec{t} \wedge \psi(s)$, where $\vec{t}$ is a vector of ground terms, and $\psi(s)$ is a formula whose only free variable is s. We denote by $\text{LE}(\alpha)$ the set of all fluents on which α has local effects.

Definition 4.5 We say that α is normal if for each fluent F, all the fluents that appear in γ_F^+ and γ_F^- are in $\text{LE}(\alpha)$.

Clearly, both context-free and local-effect actions are normal actions.

Definition 4.6 We say that $\mathcal{D}_{S_0}$ is normal wrt α if for each fluent $F \notin \text{LE}(\alpha)$, $\mathcal{D}_{S_0}$ is semi-definitional wrt F.

Thus any fluent $F \in \text{LE}(\alpha)$ can appear in $\mathcal{D}_{S_0}$ in an arbitrary way. We now have the main result of this section:

Theorem 4.7 *Let $\mathcal{D}_{S_0}$ be normal wrt a normal action α. Then progression of $\mathcal{D}_{S_0}$ wrt α is FO definable and computable.*

Proof: By Theorem 2.10, we need to forget the lifting predicates in $\bigwedge(\mathcal{D}_{una} \cup \mathcal{D}_{S_0} \cup \mathcal{D}_{ss}[\alpha, S_0]) \uparrow S_0$. Since α is normal, for each fluent F, all the fluents that appear in γ_F^+ and γ_F^- are in $\text{LE}(\alpha)$. By Proposition 4.3, for each $F \notin \text{LE}(\alpha)$, $\mathcal{D}_{ss}[\alpha, S_0]$ is semi-definitional wrt $F(\vec{x}, S_0)$. Since $\mathcal{D}_{S_0}$ is normal wrt α, for each fluent $F \notin \text{LE}(\alpha)$, $\mathcal{D}_{S_0}$ is semi-definitional wrt F. By applying Theorem 4.2, we forget the lifting predicate for $F \notin \text{LE}(\alpha)$. Now by applying Theorem 3.6, we forget the lifting predicate for $F \in \text{LE}(\alpha)$. ■

Example 4.1 The following is $\mathcal{D}_{ss}$ for the briefcase domain:
$at(x, l, do(a, s)) \equiv (\exists b)[a = move(b, l) \wedge (x = b \vee in(x, b, s))] \vee$
$\quad at(x, l, s) \wedge \neg(\exists b, m)[a = move(b, m) \wedge (x = b \vee in(x, b, s))]$,
$in(x, b, do(a, s)) \equiv a = putin(x, b) \vee$
$\quad in(x, b, s) \wedge \neg a = getout(x, b)$.

For a ground action $\alpha = move(c_1, c_2)$, by using $\mathcal{D}_{una}$, $\mathcal{D}_{ss}[\alpha, S_0]$ can be simplified as follows:
$at(x, l, do(\alpha, S_0)) \equiv l = c_2 \wedge (x = c_1 \vee in(x, c_1, S_0)) \vee$
$\quad at(x, l, S_0) \wedge \neg(x = c_1 \vee in(x, c_1, S_0))$,
$in(x, b, do(\alpha, S_0)) \equiv in(x, b, S_0)$.
Clearly, α has local effects on in, and it is a normal action.

Now let $\mathcal{D}_{S_0}$ be as follows:
$\exists x \forall y \neg in(x, y, S_0)$, $\neg in(A_1, B_1, S_0)$,
$in(A_2, B_2, S_0) \vee in(A_2, B_3, S_0)$,
$at(b, l, S_0) \wedge in(x, b, S_0) \supset at(x, l, S_0)$,
$at(x, l', S_0) \wedge l \neq l' \supset \neg at(x, l, S_0)$,
$b = B_1 \wedge l = L_1 \vee b = B_2 \wedge l = L_2 \supset at(x, l, S_0)$,
$b = B_3 \wedge (l = L_1 \vee l = L_2) \supset \neg at(x, l, S_0)$.
Then $\mathcal{D}_{S_0}$ is normal wrt $\alpha = move(B_1, L_2)$. To progress it wrt α, we first apply Theorem 4.2 to forget the lifting predicate for $at(x, l, s)$, and obtain a set Σ of sentences as follows:

1. If $\phi \in \mathcal{D}_{S_0}$ does not mention fluent at, then $\phi \in \Sigma$.
2. Add to Σ the following sentences:
   $l = L_2 \wedge (x = B_1 \vee in(x, B_1, S_0)) \supset at(x, l, S_\alpha)$,
   $l \neq L_2 \wedge (x = B_1 \vee in(x, B_1, S_0)) \supset \neg at(x, l, S_\alpha)$.
3. If $\phi \supset at(x, l, S_0)$ is in $\mathcal{D}_{S_0}$, then add to Σ the sentence
   $\phi \wedge \neg(x = B_1 \vee in(x, B_1, S_0)) \supset at(x, l, S_\alpha)$.
4. If $\phi \supset \neg at(x, l, S_0)$ is in $\mathcal{D}_{S_0}$, add to Σ the sentence
   $\phi \wedge (l \neq L_2 \vee x \neq B_1 \wedge \neg in(x, B_1, S_0)) \supset \neg at(x, l, S_\alpha)$.

Now since we have $in(x, b, S_\alpha) \equiv in(x, b, S_0)$, we simply replace each occurrence of S_0 in Σ with S_α; the result together with $\mathcal{D}_{una}$ is a progression of $\mathcal{D}_{S_0}$ wrt α.

5 Progression of proper$^+$ KBs

In Sections 3 and 4, we showed that for local-effect and normal actions, progression is FO definable and computable. However, the progression may not be efficiently computable.

In this section, we show that for local-effect and normal actions, progression is not only FO definable but also efficiently computable under the two constraints that the initial KB is in the form of the so-called proper$^+$ KBs, which represent first-order disjunctive information, and the successor state axioms are essentially quantifier-free.

Proper$^+$ KBs were proposed by Lakemeyer and Levesque [2002] as a generalization of proper KBs, which were proposed by Levesque [1998] as an extension of databases. Intuitively, a proper$^+$ KB is equivalent to a (possibly infinite) set of ground clauses. A tractable limited reasoning service has been developed for proper$^+$ KBs [Liu *et al.*, 2004; Liu and Levesque, 2005]. What is particularly interesting about our results here is that progression of proper$^+$ KBs is definable as proper$^+$ KBs, so that we can make use of the available tractable reasoning service.

To formally define proper$^+$ KBs, we use a FO language $\mathcal{L}_c$ with equality, a countably infinite set of constants, which are intended to be unique names, and no other function symbols. We let e range over ewffs, *i.e.*, quantifier-free formulas whose only predicate is equality. We denote by $\mathcal{E}$ the axioms of equality and the set of formulas $\{(c \neq c') \mid c \text{ and } c' \text{ are distinct constants}\}$. We let $\forall\phi$ denote the universal closure of ϕ.

Definition 5.1 Let e be an ewff and d a clause. Then a formula of the form $\forall(e \supset d)$ is called a $\forall$-clause. A KB is called *proper*$^+$ if it is a finite non-empty set of $\forall$-clauses.

Example 5.1 Consider our blocks world. The following is a initial KB $\mathcal{D}_{S_0}$ which is proper$^+$:
$on(x, y, S_0) \supset \neg clear(y, S_0)$,
$on(x, y, S_0) \wedge eh(y, S_0) \supset \neg eh(x, S_0)$,
$x = A \vee x = C \supset clear(x, S_0)$,
$x = D \vee x = E \vee x = F \supset \neg eh(x, S_0)$,
$x = A \wedge y = B \vee x = B \wedge y = F \supset on(x, y, S_0)$,
$on(C, D, S_0) \vee on(C, E, S_0)$.

We begin with forgetting in proper$^+$ KBs. We first introduce some definitions and propositions.

Definition 5.2 Let ϕ be a sentence, and p a ground atom. We say that p is irrelevant to ϕ if forget$(\phi, p) \Leftrightarrow \phi$.

Proposition 5.3 *Let p be a ground atom. Let ϕ_1, ϕ_2, ϕ_3 be sentences such that p is irrelevant to them. Then forget$((\phi_1 \supset p) \wedge (p \supset \phi_2) \wedge \phi_3, p) \Leftrightarrow (\phi_1 \supset \phi_2) \wedge \phi_3$.*

Proposition 5.4 *Let $\phi = \forall(e \supset d)$ be a $\forall$-clause, and $P(\vec{c})$ a ground atom. Suppose that for any $P(\vec{t})$ appearing in d, $e \wedge \vec{t} = \vec{c}$ is unsatisfiable. Then $P(\vec{c})$ is irrelevant to ϕ.*

Definition 5.5 Let Σ be a proper$^+$ KB, and $P(\vec{c})$ a ground atom. We say that Σ is in normal form wrt $P(\vec{c})$, if for any $\forall(e \supset d) \in \Sigma$, and for any $P(\vec{t})$ appearing in d, either $\vec{t}$ is $\vec{c}$ or $e \wedge \vec{t} = \vec{c}$ is unsatisfiable.

Proposition 5.6 *Let $P(\vec{c})$ be a ground atom. Then every proper$^+$ KB can be converted into an equivalent one which is in normal form wrt $P(\vec{c})$. This can be done in $O(n+2^w m)$ time, where n is the size of Σ, m is the size of sentences in Σ where P appears, and w is the maximum number of appearances of P in a sentence of Σ.*

Proof: Let $\phi = \forall(e \supset d)$ be a $\forall$-clause. Let $P(\vec{t_1}), \ldots, P(\vec{t_k})$ be all the appearances of P in ϕ, and let $\Theta = \{\bigwedge_{i=1}^{k} \vec{t_i} \circ_i \vec{c} \mid \circ_i \in \{=, \neq\}\}$. Let $\theta \in \Theta$. We let $d[\theta]$ denote d with each $P(\vec{t_i})$, $1 \leq i \leq k$, replaced by $P(\vec{c})$ if θ contains $\vec{t_i} = \vec{c}$. We use $\phi[\theta]$ to denote $\forall(e \wedge \theta \supset d[\theta])$. Obviously, ϕ is equivalent to the theory $\{\phi[\theta] \mid \theta \in \Theta\}$, which we denote by NF$(\phi, P(\vec{c}))$. For a proper$^+$ KB Σ, we convert it into the union of NF$(\phi, P(\vec{c}))$ where $\phi \in \Sigma$. ■

In the above proof, we can remove those generated $\forall$-clauses $\forall(e \supset d)$ where d contains complemental literals or e is unsatisfiable wrt $\mathcal{E}$. For example, the ewff $x = y \wedge x = A \wedge y = B$ is unsatisfiable wrt $\mathcal{E}$. Also, a $\forall$-clause $\forall(e \wedge t = c \supset d)$ can be simplified to $\forall(e \supset d)(t/c)$. Finally, an ewff can be simplified by use of $\mathcal{E}$.

Definition 5.7 Let $\phi_1 = \forall(e_1 \supset d_1 \vee P(\vec{t}))$ and $\phi_2 = \forall(e_2 \supset d_2 \vee \neg P(\vec{t}))$ be two $\forall$-clauses, where $\vec{t}$ is a vector of constants or a vector of distinct variables. Without loss of generality, we assume that ϕ_1 and ϕ_2 do not share variables other than those contained in $\vec{t}$. We call the $\forall$-clause $\forall(e_1 \wedge e_2 \supset d_1 \vee d_2)$ the $\forall$-resolvent of the two input clauses wrt $P(\vec{t})$.

Theorem 5.8 *Let Σ be a proper$^+$ KB, and $P(\vec{c})$ a ground atom. Then the result of forgetting $P(\vec{c})$ in Σ is definable as a proper$^+$ KB and can be computed in $O(n + 4^w m^2)$ time, where n, w, and m are as above.*

Proof: We first convert Σ into normal form wrt $P(\vec{c})$. Then we compute all $\forall$-resolvents wrt $P(\vec{c})$ and remove all clauses with $P(\vec{c})$. This results in a proper$^+$ KB, which, by Propositions 5.3 and 5.4, is a result of forgetting $P(\vec{c})$ in Σ. ■

Theorem 5.9 *Let Σ be a proper$^+$ KB which is semi-definitional wrt predicate P. Then the result of forgetting P in Σ is definable as a proper$^+$ KB and can be computed in $O(n + m^2)$ time, where n is the size of Σ, and m is the size of sentences in Σ where P appears.*

Proof: We compute all $\forall$-resolvents wrt $P(\vec{x})$ and remove all clauses containing $P(\vec{x})$. This results in a proper$^+$ KB, which, by Theorem 4.2, is a result of forgetting P in Σ. ■

In the above theorems (Theorems 5.8 and 5.9), it is reasonable to assume that $w = O(1)$ and $m^2 = O(n)$. Under this assumption, both forgetting can be computed in $O(n)$ time.

Based on the above theorems, we have the following results concerning progression of proper$^+$ KBs. We first introduce a constraint on successor state axioms.

Definition 5.10 An SSA is essentially quantifier-free if for each ground action α, by using $\mathcal{D}_{una}$, each of $\gamma_F^+(\vec{x}, \alpha, s)$ and $\gamma_F^-(\vec{x}, \alpha, s)$ can be simplified to a quantifier-free formula.

For example, the SSAs for our blocks world and briefcase examples are essentially quantifier-free. For a local-effect SSA, if each context-formula is quantifier-free, then it is essentially quantifier-free. In general, if both $\gamma_F^+(\vec{x}, a, s)$ and $\gamma_F^-(\vec{x}, a, s)$ are disjunctions of formulas of the form $\exists\vec{z}[a = A(\vec{u}) \wedge \phi(\vec{x}, \vec{z}, s)]$, where $\vec{u}$ contains $\vec{z}$, and ϕ is quantifier-free, then the SSA is essentially quantifier-free.

Proposition 5.11 *Suppose $\mathcal{D}_{ss}$ is essentially quantifier-free. Then $\mathcal{D}_{ss}[\alpha, S_0]$ is definable as a proper$^+$ KB.*

Theorem 5.12 *Suppose that $\mathcal{D}$ is local-effect, $\mathcal{D}_{ss}$ is essentially quantifier-free, and $\mathcal{D}_{S_0}$ is proper$^+$. Then progression of $\mathcal{D}_{S_0}$ wrt any ground action α is definable as a proper$^+$ KB and can be efficiently computed.*

Theorem 5.13 *Suppose that $\mathcal{D}_{ss}$ is essentially quantifier-free, α is a normal action, and $\mathcal{D}_{S_0}$ is a proper$^+$ KB which is normal wrt α. Then progression of $\mathcal{D}_{S_0}$ wrt α is definable as a proper$^+$ KB and can be efficiently computed.*

Example 5.1 continued. We now progress $\mathcal{D}_{S_0}$ wrt $\alpha = move(A, B, C)$. For simplicity, we remove the second sentence from $\mathcal{D}_{S_0}$. The characteristic set of α is $\Omega = \{clear(B, s), clear(C, s), on(A, B, s), on(A, C, s), eh(A, s)\}$. $\mathcal{D}_{ss}[\Omega]$ is simplified to the following proper$^+$ KB: $\{clear(B, S_\alpha), \neg clear(C, S_\alpha), \neg on(A, B, S_\alpha), on(A, C, S_\alpha), eh(A, S_\alpha) \vee eh(C, S_0), \neg eh(A, S_\alpha) \vee \neg eh(C, S_0)\}$.

We convert $\mathcal{D}_{S_0}$ into normal form wrt $\Omega(S_0)$, and obtain after simplification:

$y \neq B \wedge y \neq C \wedge (x \neq A \vee y \neq B \wedge y \neq C) \supset (on(x, y, S_0) \supset \neg clear(y, S_0))$,
$x \neq A \supset (on(x, B, S_0) \supset \neg clear(B, S_0))$,
$x \neq A \supset (on(x, C, S_0) \supset \neg clear(C, S_0))$,
$on(A, B, S_0) \supset \neg clear(B, S_0)$,
$on(A, C, S_0) \supset \neg clear(C, S_0)$,
$x = D \vee x = E \vee x = F \supset \neg eh(x, S_0)$,
$clear(A, S_0), clear(C, S_0), on(A, B, S_0), on(B, F, S_0)$,
$on(C, D, S_0) \vee on(C, E, S_0)$.

We now do resolution on $\mathcal{D}_{S_0} \cup \mathcal{D}_{ss}[\Omega]$ wrt atoms in $\Omega(S_0)$, delete all clauses with some atom from $\Omega(S_0)$, and obtain the following set Σ:

$clear(B, S_\alpha), \neg clear(C, S_\alpha), \neg on(A, B, S_\alpha), on(A, C, S_\alpha)$,
$eh(A, S_\alpha) \vee eh(C, S_0), \neg eh(A, S_\alpha) \vee \neg eh(C, S_0)$,
$y \neq B \wedge y \neq C \wedge (x \neq A \vee y \neq B \wedge y \neq C) \supset (on(x, y, S_0) \supset \neg clear(y, S_0))$,
$x \neq A \supset \neg on(x, C, S_0)$,
$x = D \vee x = E \vee x = F \supset \neg eh(x, S_0)$,
$clear(A, S_0), on(B, F, S_0), on(C, D, S_0) \vee on(C, E, S_0)$.

We replace every occurrence of S_0 in Σ with S_α; the result together with $\mathcal{D}_{una}$ is a progression of $\mathcal{D}_{S_0}$ wrt α.

6 Conclusions

In this paper, we have presented the following results. First, we showed that for local-effect actions, progression is FO definable and computable. This result is stronger than the one obtained by Vassos *et al.* [2008], and our proof is a very simple one via the concept of forgetting. Next, we went beyond local-effect actions, and showed that for normal actions, *i.e.*, actions whose effects do not depend on the fluents on which the actions have non-local effects, if the initial KB is semi-definitional wrt these fluents, progression is FO definable and computable. Third, we showed that for local-effect actions whose successor state axioms are essentially quantifier-free, progression of proper$^+$ KBs is definable as proper$^+$ KBs and can be efficiently computed. Thus we can utilize the available tractable limited reasoning service for proper$^+$ KBs. As an extension of our first result, we have shown that for finite-effect actions, which change the truth values of fluents at only a finite number of instances, progression is FO definable. We have also shown that in the presence of functional fluents, our first and second results still hold. For lack of space, these results will be presented in a longer version of the paper. For the future, we would like to implement a Golog interpreter based on progression of proper$^+$ KBs, which we expect will lead to a more efficient version of Golog compared to the current implementation based on regression.

Acknowledgments

We thank the anonymous reviewers for very helpful comments. This work was supported in part by the European Union Erasmus Mundus programme.

References

[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.

[Gelfond and Lifschitz, 1993] M. Gelfond and V. Lifschitz. Representing actions and change by logic programs. *Journal of Logic Programming*, 17(2–4):301–323, 1993.

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

[Levesque *et al.*, 1997] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. Scherl. Golog: A logic programming language for dynamic domains. *J. of Logic Programming*, 31:59–84, 1997.

[Levesque, 1998] H. J. Levesque. A completeness result for reasoning with incomplete first-order knowledge bases. In *Proc. KR-98*, 1998.

[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 Levesque, 2005] Y. Liu and H. J. Levesque. Tractable reasoning in first-order knowledge bases with disjunctive information. In *Proc. AAAI-05*, 2005.

[Liu *et al.*, 2004] Y. Liu, G. Lakemeyer, and H. J. Levesque. A logic of limited belief for reasoning with disjunctive information. In *Proc. KR-04*, 2004.

[Nonnengart *et al.*, 1999] A. Nonnengart, H. J. Ohlbach, and A. Szalas. Elimination of predicate quantifiers. In *Logic, Language and Reasoning, Part I*, pages 159–181. 1999.

[Reiter, 2001] R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. 2001.

[Vassos and Levesque, 2008] S. Vassos and H. J. Levesque. On the progression of situation calculus basic action theories: Resolving a 10-year-old conjecture. In *Proc. AAAI-08*, 2008.

[Vassos *et al.*, 2008] S. Vassos, G. Lakemeyer, and H. J. Levesque. First-order strong progression for local-effect basic action theories. In *Proc. KR-08*, 2008.