Progressing Basic Action Theories with Non-Local Effect Actions

Stavros Vassos[1], Sebastian Sardina[2], and Hector J. Levesque[1]

[1] Department of Computer Science, University of Toronto, Toronto, Canada
{stavros,hector}@cs.toronto.edu
[2] School of Computer Science, RMIT University, Melbourne, Australia
ssardina@cs.rmit.edu.au

Abstract. In this paper we propose a practical extension to some recent work on the progression of action theories in the situation calculus. Based on the notion of *safe-range queries* from database theory and *just-in-time* action histories, we present a new type of action theory, called *range-restricted*, that allows actions to have non-local effects with a restricted range. These theories can represent incomplete information in terms of *possible closures* for fluents and can be progressed by directly updating the database in an algorithmic manner.

1 Introduction

One of the requirements for building agents with a pro-active behavior is the ability to reason about action and change. The ability to *predict* how the world will be after performing a sequence of actions is the basis for offline automated planning, scheduling, web-service composition, etc. In the situation calculus [1] such reasoning problems are examined in the context of the basic action theories (BATs), i.e., logical theories that specify the preconditions and effects of actions, and an initial knowedge base (KB) that represents the initial state of the world.

A BAT can be used to solve offline problems as well as to equip a situated agent with the ability to *keep track* of the current state of the world. In those cases, it is mandatory that the BAT be (periodically) updated so that the initial KB is replaced by a new one reflecting the changes due to the actions that have already occurred. This is identified as the problem of *progression* for BATs [2]. Unfortunately, the updated KB requires second-order logic in the general case [2]. For this reason, restrictions on the BATs have been proposed so that the updated KB is first-order representable. It was recently shown that progression is practical provided actions are limited to have local effects only [3].

The local-effect restriction essentially means that all the properties of the world that may be affected by an action are directly specified by the arguments of the action. This assumption is too restrictive for many realistic scenarios. For instance, the action of moving a container which causes all objects in it to be moved as well cannot be represented. In this paper, we extend local-effect BATs to account for actions that are non-local but have a restricted range. Based on the same techniques that are used in [3] we describe a method for progression such that the new KB is first-order and finite.

2 Formal preliminaries

The situation calculus [1] is a first-order logic language with some limited second-order features, designed for representing and reasoning about dynamically changing worlds. The constant S_0 is used to denote the initial situation where no action has been performed and sequences of actions are built using the function *do* so that $do(a, s)$ denotes the situation resulting from performing action a in situation s. Relations whose truth value varies from situation to situation are called *fluents* and are denoted by predicate symbols taking a situation term as their last argument. The predicate $Poss(a, s)$ is used to state that action a is executable in situation s. Finally, often we will focus on sentences that refer to a particular situation. For this purpose, for any σ, we define the set of *uniform* formulas in σ to be all those formulas that do not mention any other situation terms except for σ, do not mention *Poss*, and where σ is not used by any quantifier [2].

Within the language we can formulate theories that describe how the world changes as the result of the available actions. We focus on the *basic action theories (BATs)* [1] of the form $\mathcal{D} = \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una} \cup \mathcal{D}_0 \cup \mathcal{D}_{fnd}$, where:

1. $\mathcal{D}_{ap}$ is the set of action precondition axioms (PAs), one per action symbol A, of the form $Poss(A(\boldsymbol{y}), s) \equiv \Pi_A(\boldsymbol{y}, s)$, where $\Pi_A(\boldsymbol{y}, s)$ is uniform in s.
2. $\mathcal{D}_{ss}$ is the set of successor state axioms (SSAs), one per fluent symbol F, of the form $F(\boldsymbol{x}, do(a, s)) \equiv \gamma^+_F(\boldsymbol{x}, a, s) \vee (F(\boldsymbol{x}, s) \wedge \neg\gamma^-_F(\boldsymbol{x}, a, s))$, where $\gamma^+_F(\boldsymbol{x}, a, s)$ is uniform in s and captures the positive effects of actions. Similarly, $\gamma^-_F(\boldsymbol{x}, a, s)$ captures the negative effects.
3. $\mathcal{D}_{una}$ is the set of unique-names axioms for actions.
4. $\mathcal{D}_0$, the *initial knowledge base (KB)*, is a set of sentences uniform in S_0.
5. $\mathcal{D}_{fnd}$ is a set of domain independent axioms that define the legal situations.

We follow the definition of a *strong progression* [3] that is based on the second-order sentence $Pro(\alpha, \mathcal{D})$. In particular, a set $\mathcal{D}_\alpha$ is a progression of a finite $\mathcal{D}_0$ wrt α and $\mathcal{D}$ iff it is logically equivalent to $Pro(\alpha, \mathcal{D})$. Although a strong progression is defined in second-order logic we are interested in cases where we can find a $\mathcal{D}_\alpha$ that is *first-order representable*. In the sequel, we shall present a restriction on $\mathcal{D}$ that is a sufficient condition for doing this.

3 Range-restricted basic action theories

We present a new type of basic action theories such that $\mathcal{D}_0$ is a *database of possible closures* and the SSAs in $\mathcal{D}_{ss}$ are built on *range-restricted* formulas.

3.1 A database of possible closures

We treat each fluent $F(\boldsymbol{x}, w, s)$ as a *multi-valued function*, where the w is considered as the "*output*" and $\boldsymbol{x}$ as the "*input*" of the function. This distinction is important as we require that $\mathcal{D}_0$ contains sentences expressing disjunctive information only about the output of fluents.

Definition 1. *Let $V = \{e_1, \ldots, e_m\}$ be a set of constants and τ a fluent atom of the form $F(\boldsymbol{c}, w, S_0)$, where $\boldsymbol{c}$ is a vector of constants. The (atomic)* closure of τ on $\{e_1, \ldots, e_m\}$ *is the sentence:*

$$\forall w.F(\boldsymbol{c}, w, S_0) \equiv (w = e_1 \vee \cdots \vee w = e_m).$$

The notion generalizes to the vector $\boldsymbol{\tau}$ of atoms of this form and the vector of sets of constants $\boldsymbol{V}$, as the conjunction of each of the atomic closures of τ_i on V_i. A possible closures axiom (PCA) *for $\boldsymbol{\tau}$ is a disjunction of closures of $\boldsymbol{\tau}$. We say that each disjunct of the PCA is a* possible closure *of $\boldsymbol{\tau}$.*

A closure of $\boldsymbol{\tau}$ expresses complete information while a PCA for $\boldsymbol{\tau}$ expresses disjunctive information about $\boldsymbol{\tau}$. For example, let $Near(x, y, s)$ represent that y is near the object x, and χ_1 be $\forall w.Near(bomb, w, S_0) \equiv (w = agent \vee w = box_1)$. Then, χ_1 is the atomic closure of $Near(bomb, w, S_0)$ on $\{agent, box_1\}$ which along with the unique-names axioms for constants implies that there are *exactly* two objects near the bomb, namely *agent*) and box_1. Similarly, let χ_2 be the closure of $Near(bomb, w, S_0)$ on $\{agent, box_2\}$. Then, $\chi_1 \vee \chi_2$ is a PCA for $Near(bomb, w, S_0)$ expressing that there are exactly two objects near the bomb, one being the agent and the other being either box_1 or box_2.

We say that an initial database $\mathcal{D}_0$ is a *database of possible closures (DBPC)* iff it is a finite set of PCAs uch that there is no fluent atom F with ground input $\boldsymbol{c}$ that appears in more than one PCA. Our method for progressing a DBPC is based on computing the *possible answers* to certain formulas in $\mathcal{D}_{ss}$.

Definition 2. *Let $\mathcal{D}_0$ be a DBPC and $\gamma(\boldsymbol{x})$ a formula uniform in S_0. The* possible answers *of γ wrt $\mathcal{D}_0$, denoted as $\mathsf{pans}(\gamma, \mathcal{D}_0)$, is the smallest set of pairs $(\boldsymbol{c}, \chi)$ such that:*

- *χ is a closure of some vector $\boldsymbol{\tau}$ such that $\mathcal{E} \cup \{\chi\} \models \gamma(\boldsymbol{c})$, where $\mathcal{E}$ is the set of unique-names axioms for objects in $\mathcal{L}$;*
- *χ is consistent with $\mathcal{D}_0$ and minimal in the sense that every atomic closure in χ is necessary.*

Intuitively, $\mathsf{pans}(\gamma, \mathcal{D}_0)$ characterizes all the cases where the formula $\gamma(\boldsymbol{x})$ is satisfied in a model of $\mathcal{D}_0$ for some instantiation of $\boldsymbol{x}$. For example, let $\mathcal{D}_0$ be $\chi_1 \vee \chi_2$ and $\gamma(x)$ the query $Near(bomb, x, S_0)$. Then, $\mathsf{pans}(\gamma, \mathcal{D}_0)$ is the set

$$\{(agent, \chi_1), (box_1, \chi_1), (agent, \chi_2), (box_2, \chi_2)\}.$$

It is important to observe is that possible answers may be *infinite*. We now proceed to specify two conditions that ensure that $\mathsf{pans}(\gamma, \mathcal{D}_0)$ is a finite set.

3.2 Range-restricted and just-in-time formulas

We distinguish two ways that the possible answers can be infinite. Let $\mathcal{D}_0$ be $\chi_1 \vee \chi_2$. In the case of $\mathsf{pans}(Near(agent, x, S_0), \mathcal{D}_0)$ this happens because what is being queried is *completely* unknown in $\mathcal{D}_0$. In the case of $\mathsf{pans}(\neg Near(bomb, w, S_0), \mathcal{D}_0)$ though, this happens because the DBPC is queried for negative information. The first case can be avoided when the formula is *just-in-time* in the following sense:

Definition 3. *Let $\mathcal{D}_0$ be a DBPC. A formula γ is* just-in-time (JIT) *wrt $\mathcal{D}_0$ iff for all $(\boldsymbol{c}, \chi)$ in $\mathsf{pans}(\gamma, \mathcal{D}_0)$, χ consists of atomic closures from $\mathcal{D}_0$.*

This essentially means that all the possible answers to γ rely on fluent atoms that are mentioned in the DBPC. In order to avoid an infinite set of possible answers we also need to ensure that it is *range-restricted* in the following sense:

Definition 4. *The formula γ uniform in S_0 is* safe-range *wrt a set of variables X according to the following rules:*

1. *for the atomic case let $\boldsymbol{t}$ be a vector of variables and constants:*
   - *$x = c$ is safe-range wrt $\{x\}$;*
   - *$F(\boldsymbol{t}, c, S_0)$ is safe-rage wrt $\{\}$, $F(\boldsymbol{t}, x, S_0)$ is safe-range wrt $\{x\}$ provided that $x \notin \boldsymbol{t}$, and safe-range wrt $\{\}$ otherwise;*
2. *if ϕ is safe-range wrt X_ϕ, ψ is safe-range wrt X_ψ then,*
   - *$\phi \vee \psi$ and $\phi \wedge \psi$ are safe-range wrt $X_\phi \cap X_\psi$ and $X_\phi \cup X_\psi$ respectively;*
   - *$\neg\phi$ is safe-range wrt $\{\}$;*
   - *$\exists x\phi$ is safe-range wrt $X_\phi/\{x\}$ provided that $x \in X_\phi$;*
3. *no other formula is safe-range.*

A formula is range-restricted *iff it is safe-range wrt the set of its free variables.*

For example, $\mathit{Near}(x, y, S_0)$ is safe-range wrt $\{y\}$, but not range-restricted and not JIT wrt the $\mathcal{D}_0$ of our example. On the other hand, $\mathit{Near}(\mathit{bomb}, y, S_0)$ and $\mathit{Near}(\mathit{bomb}, y, S_0) \wedge \mathit{Status}(y, z, S_0)$ are range-restricted as well as JIT wrt $\mathcal{D}_0$. We now state our main result which forms the basis of our progression mechanism.

Theorem 1. *Let $\mathcal{D}_0$ be a DBPC, and $\gamma(\boldsymbol{x})$ be a formula uniform in S_0 that is range-restricted and just-in-time wrt $\mathcal{D}_0$. Then, $\mathsf{pans}(\gamma, \mathcal{D}_0)$ is a* finite *set $\{(\boldsymbol{c}_1, \chi_1), \ldots, (\boldsymbol{c}_n, \chi_n)\}$ such that the following holds:*

$$\mathcal{D}_0 \cup \mathcal{E} \models \forall \boldsymbol{x}.\gamma(\boldsymbol{x}) \equiv \bigvee_{i=1}^{n} (\boldsymbol{x} = \boldsymbol{c}_i \wedge \chi_i).$$

Now let γ be the formula $\gamma_F^+(\boldsymbol{x}, \alpha, S_0)$ of some SSA in $\mathcal{D}_{ss}$, instantiated for the ground action α and the initial situation S_0. Note that this result is similar to the one in [3] that specifies the *context set* of α and forms the basis for a progression method. The idea then is to build SSAs from range-restricted formulas and allow progression to take place only when the JIT assumption also holds.

3.3 Progression

A BAT is *range-restricted* iff it includes the set $\mathcal{E}$ of the unique-names axioms for objects, $\mathcal{D}_0$ is a database of possible closures, and for all F, the formulas $\gamma_F^+(\boldsymbol{x}, a, s)$ and $\gamma_F^-(\boldsymbol{x}, a, s)$ are disjunctions of the form:

$$\exists \boldsymbol{z}(a = A(\boldsymbol{y}) \wedge \phi(\boldsymbol{x}, \boldsymbol{z}, s)),$$

where $\boldsymbol{y}$ may contain some variables from $\boldsymbol{x}$, $\boldsymbol{z}$ corresponds to the remaining variables of $\boldsymbol{y}$, and the formula ϕ is such that $\phi(\boldsymbol{x}, \boldsymbol{z}, S_0)$ is safe-range wrt the variables in $\boldsymbol{x}$. It follows that whenever $\gamma_F^+(\boldsymbol{x}, \alpha, S_0)$ and $\gamma_F^-(\boldsymbol{x}, \alpha, S_0)$ in $\mathcal{D}_{ss}$ are also *just-in-time* wrt $\mathcal{D}_0$ then there is a finite set of ground fluent atoms that may be affected and we can progress $\mathcal{D}_0$ by using the same technique as in [3].

4 Related and future work

The notion of progression for BATs was first introduced by Lin and Reiter [2]. The version we use here is due to Vassos *et al* [3] which we extended slightly to account for sensing. Liu and Levesque [4] introduced the *local-effect* assumption for actions when they proposed a weaker version of progression that is logically incomplete but remains practical. Vassos *et al.* [3] later showed that under this assumption a correct first-order progression can be computed by updating a finite $\mathcal{D}_0$. Our restriction of a range-restricted BAT is similar. The main difference is that we do not require that the arguments $\boldsymbol{x}$ of the fluent F are included in the arguments $\boldsymbol{y}$ of the action, thus handling cases like the bomb example. To stay practical though we had to restrict the structure of $\mathcal{D}_0$.

The notions of the safe-range and range-restricted queries come from the database theory where this form of "safe" queries has been extensively studied [5]. The notion of just-in-time formulas was introduced for a different setting in [6] and, in our case, is also related to the *active domain* of a database [5]. Outside of the situation calculus, Thielscher [7] defined a dual representation for BATs based on state update axioms that explicitly define the direct effects of each action, and investigated progression in this setting.

5 Conclusions

In this paper, we proposed a new type of basic action theories, where the initial description is a set of *possible closures* and the effects of actions have a *restricted range*. For these theories we presented a method that computes a finite first-order progression by directly updating the initial database. To the best of our knowledge, it is the first result on the progression of basic action theories with an infinite domain, and incomplete information that goes beyond the local-effect assumption.

References

1. Reiter, R.: Knowledge in Action. Logical Foundations for Specifying and Implementing Dyn. Sys. MIT Press (2001)
2. Lin, F., Reiter, R.: How to progress a database. Art. Int. **92**(1-2) (1997) 131–167
3. Vassos, S., Lakemeyer, G., Levesque, H.J.: First-order strong progression for local-effect basic action theories. In: Proc. of KR. (2008) 662–272
4. Liu, Y., Levesque, H.J.: Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In: Proc. of IJCAI. (2005)
5. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases : The Logical Level. Addison Wesley (1994)
6. De Giacomo, G., Levesque, H.J., Sardina, S.: Incremental execution of guarded theories. Computational Logic **2**(4) (2001) 495–525
7. Thielscher, M.: From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. Art. Int. **111**(1-2) (1999) 277–299