CHALLENGES FOR ACTION THEORIES (EXTENDED ABSTRACT)

M. THIELSCHER
Department of Computer Science, Technical University Darmstadt, Alexanderstr 10, 64283 Darmstadt (Germany)

Action theories serve as a formalism to represent and reason about domains in which the execution of actions plays a central role. This is the case, for example, if one wants to model an agent who interacts with its environment, i.e., the part of the world it is able to affect. Most importantly, an *autonomous* agent needs precise knowledge as to the effects of its actions in order to act purpose-oriented and so to achieve pre-determined goals. The latter requires to draw the right conclusions from this knowledge in view of particular situations, in which the agent has acquired partial knowledge about the current state of the environment and has a certain goal in mind. The most distinguishing aspect of action theories is that the specification of actions and their effects shall be as intuitive and natural as possible. This raises specific challenges for action theories, of which the most famous are, in historical order, the *Frame Problem*, the *Qualification Problem*, and the *Ramification Problem*.

1 Reasoning about actions

Automating commonsense reasoning about actions and their effects was among the very first issues raised in Artificial Intelligence research. McCarthy[11] advocated the belief that multifarious intelligent behavior relies on the ability to maintain a mental model of the world and to draw the right conclusions about observations and intentions. The modern research area of Cognitive Robotics follows this paradigm in that it aims at providing autonomous robots with a cognitive component in which the environment is represented by logical sentences. Automated reasoning thus enables the robot to draw the right conclusions from this representation and so to act purpose-oriented. Three fundamental challenges for formalizing actions and their effects have been revealed by three decades of research in this field: The *Frame Problem*, the *Qualification Problem*, and the *Ramification Problem*, respectively.

2 The Frame Problem

Introduced in the context of Situation Calculus,[10] the Frame Problem concerns the at first glance trivial but in fact highly problematic challenge to specify, in logic, that non-affected properties of the world state are still true after the performance of an action. To see where the difficulties lie, suppose we want to describe the action of toggling a light switch where the effect shall be that if the switch is open initially, then it is closed afterwards, and vice versa. Let $\mathtt{toggle}(x)$ be a term denoting the action of toggling switch x, and let $\mathtt{open}(x)$ be a unary predicate stating whether switch x is currently open. Such predicates, whose truth-value may change in the course of time as a consequence of performing actions, are usually referred to as *fluents*. A straightforward, naïve description of our example action is given by the implication[a]

$$\mathtt{do}(\mathtt{toggle}(x)) \supset [\neg\mathtt{open}(x) \supset \mathtt{open}(x)] \wedge [\mathtt{open}(x) \supset \neg\mathtt{open}(x)] \tag{1}$$

where $\mathtt{do}(a)$ shall indicate that action a is being performed. This specification, however, is logically inconsistent with say, $\neg\mathtt{open}(\mathtt{s}_1) \wedge \mathtt{do}(\mathtt{toggle}(\mathtt{s}_1))$, i.e., the seemingly natural formalization of a situation where switch $\mathtt{s}_1$ is open and is about to being toggled. This is why Situation Calculus introduces an additional so-called situation argument to each action and fluent, thus restricting their scopes to a particular one out of many possible situations. The effect description of Eq. (1), for instance, may accordingly be re-formulated as

$$\begin{aligned} & [\neg\mathtt{open}(x,s) \supset \mathtt{open}(x, Do(\mathtt{toggle}(x),s))] \\ \wedge\ & [\mathtt{open}(x,s) \supset \neg\mathtt{open}(x, Do(\mathtt{toggle}(x),s))] \end{aligned}$$

where term s denotes some abstract situation and $Do(\mathtt{toggle}(x),s)$ the successor situation resulting from performing $\mathtt{toggle}(x)$ in situation s. This representation technique, however, raises the problem of how to conclude that some other fluent which holds in s, say, $\mathtt{open}(\mathtt{s}_2)$, still holds in situation $Do(\mathtt{toggle}(\mathtt{s}_1),s)$. In order that this particular conclusion be granted, an additional so-called frame axiom is required, namely,

$$x \neq y \wedge \mathtt{open}(y,s) \supset \mathtt{open}(y, Do(\mathtt{toggle}(x),s))$$

Now, the Frame Problem concerns the need for a large number of these frame axioms (the *representational* aspect of the problem)[b] and the necessity to carry, one-by-one, each unchanged fluent to the next situation (the *inferential* aspect).

It took more than two decades to solve the representational aspect of the Frame Problem to the best possible extent. The approach of Reiter[15] is

[a]Throughout the paper, variables are denoted by italic lower case letters. Variables occurring free in formulas are to be taken as universally quantified.

[b]To be more precise, if n is the number of fluents and m the number of actions of a domain, then close to $m \cdot n$ frame axioms in the above style need to be introduced.

based on pure classical logic and completely avoids the specification of frame axioms. This is accomplished by combining, separately for each fluent, in a single effect axiom all possibilities of how the fluent may change to true and to false, respectively. By virtue of being bi-conditionals, these so-called successor state axioms implicitly contain sufficient information so as to also entail any *non*-change of the fluent in question. An example specification is

$$\begin{aligned}\mathtt{open}(x, Do(a,s)) \;\equiv\; & a = \mathtt{toggle}(x) \wedge \neg\mathtt{open}(x,s) \\ & \vee \neg\,[\,\mathtt{open}(x,s) \supset a = \mathtt{toggle}(x)\,]^{\,c}\end{aligned}$$

Suppose, for instance, the initial situation S_0 be specified by the conjunction $\neg\mathtt{open}(\mathtt{s}_1, S_0) \wedge \mathtt{open}(\mathtt{s}_2, S_0)$, then our successor state axiom entails both $\mathtt{open}(\mathtt{s}_1, Do(\mathtt{toggle}(\mathtt{s}_1), S_0))$ and also $\mathtt{open}(\mathtt{s}_2, Do(\mathtt{toggle}(\mathtt{s}_1), S_0))$, tacitly assuming that $\mathtt{toggle}(\mathtt{s}_1) \neq \mathtt{toggle}(\mathtt{s}_2)$. While the concept of successor state axioms perfectly solves the representational aspect of the Frame Problem, it does not at all address the inferential aspect. For it still requires, for each non-affected fluent, separate application of one of these axioms in order to conclude that the fluent keeps its truth-value in the resulting situation.

The STRIPS framework[3] was an early development in response to the inferential challenge raised by the Frame Problem. STRIPS encodes states as sets of fluents, and the performance of actions is specified operationally, namely, by removal and addition of certain fluents to these sets. Apparently, this avoids investigation of any non-affected fluent. In compensation, the operational, non-declarative nature of this approach causes the loss of both expressiveness and flexibility of logic. With the aim of regaining the latter without losing the computational merits of STRIPS, the *Linear Connection Method*, a precursor of *Linear Logic*,[5] has been developed by Bibel.[1] It employs a non-classical, resource-sensitive implication to specify the effects of actions. In so doing, the Linear Connection Method introduces a special purpose logic which thus requires both a non-standard semantics and a non-standard inference engine.

The Fluent Calculus,[7] so named by Bornscheuer and Thielscher,[2] embeds in pure classical logic the notion of resource-sensitivity to account for the dynamics of state transitions. To this end, fluents are formally treated as terms, which can be combined via the special binary function symbol "$\circ$" to constitute so-called state terms. An example is

$$\neg\mathtt{open}(\mathtt{s}_1) \circ \mathtt{open}(\mathtt{s}_2)$$

[c] The first disjunct to the right of the equivalence symbol states that the only possibility of how $\mathtt{open}(x)$ may become true is that switch x is toggled in a situation where $\mathtt{open}(x)$ is false. The second disjunct derives from the fact that the only possibility of how $\mathtt{open}(x)$ may become false is that x is toggled in a situation where $\mathtt{open}(x)$ holds.

where formally the negation symbol "$\neg$" is treated as a unary function. When encoding a world state as a state term, it is obviously irrelevant at what position a fluent literal occurs in that term. E.g., our example state term and the term $\mathtt{open}(\mathtt{s}_2) \circ \neg\mathtt{open}(\mathtt{s}_1)$ represent identical states. Moreover, double application of function $\neg$ should always neutralize. This intuition is formally captured by stipulating the following equational axioms:

$$
\begin{array}{lrcll}
\forall x, y, z. & (x \circ y) \circ z & = & x \circ (y \circ z) & \text{(associativity)} \\
\forall x, y. & x \circ y & = & y \circ x & \text{(commutativity)} \\
\forall x. & x \circ \emptyset & = & x & \text{(unit element)} \\
\forall x. & \neg\neg x & = & x & \text{(double negation)}
\end{array}
$$

where the special constant "$\emptyset$" denotes a unit element for function $\circ$, thus corresponding to the empty collection of fluents. The effects of actions may then be specified by defining a ternary predicate $Result(s, a, s')$ which shall indicate that performing action a in state s results in state s', e.g.:

$$
\begin{array}{rcl}
Result(s, \mathtt{toggle}(x), s') & \equiv & \exists z_1 \, [\, s = \neg\mathtt{open}(x) \circ z_1 \,\wedge\, s' = \mathtt{open}(x) \circ z_1 \,] \\
& & \vee\; \exists z_2 \, [\, s = \mathtt{open}(x) \circ z_2 \,\wedge\, s' = \neg\mathtt{open}(x) \circ z_2 \,]
\end{array}
$$

Let, for instance, $S_0 = \neg\mathtt{open}(\mathtt{s}_1) \circ \mathtt{open}(\mathtt{s}_2)$, then $S_0 = \neg\mathtt{open}(\mathtt{s}_1) \circ z_1$ if and only if $z_1 = \mathtt{open}(\mathtt{s}_2)$. It follows that $Result(S_0, \mathtt{toggle}(\mathtt{s}_1), S_1)$ where $S_1 = \mathtt{open}(\mathtt{s}_1) \circ \mathtt{open}(\mathtt{s}_2)$. Notice how non-affected fluent terms contained in the initial state, here: fluent $\mathtt{open}(\mathtt{s}_2)$, are automatically included in the resulting state, too. No extra frame axioms are required to this end, and all unchanged fluents are carried to the new state in one step together with the effect of the action. The representation technique of Fluent Calculus thus copes, in pure classical logic, with both the representational and the inferential aspect of the Frame Problem. Axiomatizations using the paradigm of Fluent Calculus have by now been developed for a variety of ontological aspects—e.g., non-deterministic and concurrent actions,[2] continuous change,[6] and, last not least, both the Qualification and the Ramification Problem.[16,17]

3 The Ramification Problem

The Frame Problem arises because it is generally impracticable to provide an exhaustive description defining the result of executing an action in each possible state of the world. Action specifications are therefore to be restricted to the part of the world that they affect, and all other fluents are assumed to remain stable. Yet even this approach becomes unmanageable in complex domains if one tries to put all effects into a single complete specification. Although an action may cause only a small number of *direct* changes, they in turn may initiate a long chain of *indirect* effects that can be hard to foresee. Recall, for instance, the action of toggling a switch, which in the first place causes nothing but a change of the switch's position. However, the switch may be part of an electric circuit so that say, some light bulb is turned off as a side effect, which in turn may cause someone to hurt himself in a suddenly darkened room by running against a chair that, as a consequence, falls into a television set whose implosion activates the fire alarm and so on and so forth. The task, therefore, is to design a framework to formalize action scenarios where action specifications are not assumed to completely describe all possible effects. This is the Ramification Problem.[4]

A satisfactory solution to the Ramification Problem requires the successful treatment of two major issues. First, the universal assumption of persistence of fluents needs to be appropriately weakened to the effect that it applies only to those fluents which are unaffected by the action's direct *and* indirect effects. This can be achieved by keeping the strong assumption of persistence as it stands while considering the world description obtained through its application as a mere intermediate result.[17] Indirect effects are then accommodated by further reasoning until an overall satisfactory successor state obtains. This method accounts perfectly both for rigorous persistence of unaffected fluents and for arbitrarily complex chains of indirect effects.

The second issue that needs to be addressed in the context of the Ramification Problem arises from the observation that indirect effects typically are consequences of additional, general knowledge of domain-specific dependencies among fluents—but not all such purely logical consequences correspond to indirect effects in reality.[18,9] As an example, imagine an electric circuit consisting of two switches and a light bulb serially connected so that light is on if and only if both switches are closed. This may be formally described by the logical expression $\mathtt{light} \equiv \neg\mathtt{open}(\mathtt{s}_1) \wedge \neg\mathtt{open}(\mathtt{s}_2)$. Suppose, now, the first switch is toggled in a state where $\mathtt{open}(\mathtt{s}_1)$, $\neg\mathtt{open}(\mathtt{s}_2)$, and $\neg\mathtt{light}$ hold. Then, besides the direct effect of $\mathtt{open}(\mathtt{s}_1)$ becoming false, one also expects that the light bulb turns on. This indirect effect is inspired by the formula just mentioned, which includes the implication $\neg\mathtt{open}(\mathtt{s}_1) \wedge \neg\mathtt{open}(\mathtt{s}_2) \supset \mathtt{light}$. However, despite this being the intuitively expected result, the mere knowledge of the relationship between the switches and the bulb is not sufficient. For the above formula, $\mathtt{light} \equiv \neg\mathtt{open}(\mathtt{s}_1) \wedge \neg\mathtt{open}(\mathtt{s}_2)$, also entails the implication $\neg\mathtt{open}(\mathtt{s}_1) \wedge \neg\mathtt{light} \supset \mathtt{open}(\mathtt{s}_2)$, which suggests that instead of the light being turned on, the indirect effect of toggling the first switch is that the second one jumps its position—a result which is clearly counter-intuitive. Incorporating a suitable notion of causality solves this problem.[17] So-called causal relationships formalize statements like the following:

> The (direct or indirect) effect that $\mathtt{open}(\mathtt{s}_1)$ becomes true causes the indirect effect $\mathtt{light}$, provided $\neg\mathtt{open}(\mathtt{s}_2)$ holds.

Subsequent to the computation of all direct effects of an action in a particular state of the world, causal relationships are applied, one-by-one, to accommodate additional, indirect effects. In this way, the concept of causal relationships copes with both the aforementioned aspects of the Ramification Problem.

4 The Qualification Problem

Once in a while actions in daily life turn out to be unexecutable although they have been successfully performed in similar situations countless times before. These unexpected failures arise because executability of actions often depends on a multitude of additional conditions one is usually not aware of. The reason for this unawareness is that most of these conditions are so likely to be satisfied that they are simply assumed away as long as there is no evidence to the contrary. Suppose, as an example taken from daily life, someone intends to take her car for a ride. Then usually she does not first make sure that no potato in the tail pipe prevents her from starting the car, despite the fact that a clogged tail pipe necessarily renders this action impossible.

On the other hand, ignoring unlikely action disqualifications *prima facie* also means to being able to handle situations where the prior assumption of executability turns out wrong. The general challenge, therefore, is to weaken the assumption that actions are guaranteed to producing the expected effect once all specified preconditions are satisfied. Becoming an assumption *by default*, it is to be made as long as there is no evidence to the contrary. Developing a formal account of this concept within the framework of a formal action theory is the Qualification Problem.[12] Solving it is necessary in view of applying action theories to real-world environments, which do not conform with the idealistic view in that most if not all actions are potentially subject to unlikely, or *abnormal*, disqualification.

Assuming away abnormal action disqualifications by default naturally implies that if further knowledge hints at the presence of an unexpected obstacle, then one has to withdraw the previous conclusion that the action in question is qualified. Thus the entire process is intrinsically nonmonotonic. As a consequence, McCarthy proposed to employ the nonmonotonic framework of so-called circumscription,[13] with the aim of minimizing abnormal disqualifications.[14] Lifschitz,[8] however, showed that straightforward global minimization of these abnormalities is inadequate since it fails to suitably account for action disqualifications that are brought about as side effects of performing other actions—such as the deliberate introduction of a potato into the tail pipe.

Solutions to the Ramification Problem furnish a ready approach to the Qualification Problem which accommodates these abnormalities which naturally occur for reasons of causality.[16] To this end, each unlikely disqualification is *initially* assumed away, if possible, but may later occur as indirect effect of some action. For instance, a causal relationship formalizing the statement

> The (direct or indirect) effect that $\texttt{in\_pipe}(x)$ becomes true causes the indirect effect $\texttt{disqual}(\texttt{start})$.

helps us obtain a disqualification of starting the car (fluent $\texttt{disqual}(\texttt{start})$) as a side effect of inserting an object x into the tail pipe (which supposedly makes fluent $\texttt{in\_pipe}(x)$ become true). The aforementioned initial assumption of 'normality' to the largest possible extent can be obtained by means of a formal model preference criterion, which allows to distinguish those models of a scenario description which propose the least number of abnormalities.

Aside from providing means to assume away abnormal disqualifications by default while properly taking into account possible causes for these disqualifications, the successful treatment of the Qualification Problem should include the proliferation of conceivable explanations in case an action surprisingly turns out unexecutable. It may of course happen, though, that a reasoning agent is still unable to perform an action even if it has explicitly excluded, to the best of its knowledge, any imaginable preventing cause. However surprising this might be, it just proves that the agent lacks omniscience. A disqualification which is inexplicable in this sense is called miraculous. Accounting for the potential occurrence of miraculous abnormalities, too, is part of the Qualification Problem. This can be achieved by an additional minimization step, which gives rise to a refined model preference criterion.

References

1. W. Bibel. A deductive solution for plan generation. *New Generation Computing*, 4:115–132, 1986.
2. S.-E. Bornscheuer and M. Thielscher. Explicit and implicit indeterminism: Reasoning about uncertain and contradictory specifications of dynamic systems. *Journal of Logic Programming*, 31(1–3):119–155, 1997.
3. R. E. Fikes and N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

4. M. L. Ginsberg and D. E. Smith. Reasoning about action I: A possible worlds approach. *Artificial Intelligence*, 35:165–195, 1988.
5. J.-Y. Girard. Linear Logic. *Journal of Theoretical Computer Science*, 50(1):1–102, 1987.
6. C. S. Herrmann and M. Thielscher. Reasoning about continuous processes. In B. Clancey and D. Weld, ed.'s, *Proceedings of the AAAI National Conference*, pp. 639–644, Portland, OR, Aug. 1996. MIT Press.
7. S. Hölldobler and J. Schneeberger. A new deductive approach to planning. *New Generation Computing*, 8:225–244, 1990.
8. V. Lifschitz. Formal theories of action (preliminary report). In J. McDermott, ed., *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 966–972, Milan, Italy, Aug. 1987. Morgan Kaufmann.
9. V. Lifschitz. Frames in the space of situations. *Artificial Intelligence*, 46:365–376, 1990.
10. J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.
11. J. McCarthy. Programs with common sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes*, vol. 1, 77–84, London, Nov. 1958.
12. J. McCarthy. Epistemological problems of artificial intelligence. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1038–1044, Cambridge, MA, 1977. MIT Press.
13. J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.
14. J. McCarthy. Applications of circumscription to formalizing commonsense knowledge. *Artificial Intelligence*, 28:89–116, 1986.
15. R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, ed., *Artificial Intelligence and Mathematical Theory of Computation*, pp. 359–380. Academic Press, 1991.
16. M. Thielscher. Causality and the qualification problem. In L. C. Aiello *et al*, ed.'s, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pp. 51–62, Cambridge, MA, Nov. 1996. Morgan Kaufmann.
17. M. Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1–2):317–364, 1997.
18. M. Winslett. Reasoning about action using a possible models approach. In *Proceedings of the AAAI National Conference*, pp. 89–93, Saint Paul, MN, Aug. 1988.