

The Escapee Domain: A Multi-Agent Planning Domain

Gregory Gelfond

University of Nebraska Omaha
1110 S. 67th St
Omaha, Nebraska 68182

Introduction

The action language $m\mathcal{A}+$ (Gelfond 2018; Baral et al. 2015) brings together developments regarding representing and reasoning about actions and their effects from the knowledge representation and dynamic epistemic logic communities. The language’s declarative semantics is highly amenable to direct translation into a logic program under the answer-set semantics, but the reliance of modern solvers such as `clingo` (Gebser et al. 2012) on grounding makes its application limited for automating multi-agent reasoning in a natural way. In this paper, we present an axiomatization of a multi-agent domain known as the “Escapee Domain” with two purposes in mind: to serve as an introduction to the action language $m\mathcal{A}+$ itself; and also to present a challenge problem to the developers of answer-set solvers.

The Escapee Domain and the Language $m\mathcal{A}+$

Agent A is held captive by a hostile agent B . In order to escape, A must open his cell without B ’s knowledge. Fortunately, agent C is a double agent in B ’s organization and may release A from his cell. C does not want to break his cover however, so he may release A only if B is not watching. Once A has been released, he must work together with C to subdue agent B , and then make his escape. A will only work with C if he believes that C is an ally.

Representing the Domain

In $m\mathcal{A}+$, a multi-agent domain is defined over a signature $\Sigma = (\mathcal{AG}, \mathcal{F}, \mathcal{A})$ where \mathcal{AG} , \mathcal{F} , and \mathcal{A} , are finite, disjoint, non-empty sets of symbols respectively defining the *names of the agents within the domain*, the *properties of the domain* (or *fluents*), and the *elementary actions* which the agents may perform. In the interest of space, we omit the full description of the domain signature and assume that all of the relevant syntactic objects have been defined.

What is important to note is that in a multi-agent settings, fluents fall into two broad categories: *ontic fluents* and *perspective fluents*. Ontic fluents are used to describe actual physical properties of the domain, while perspective fluents, define the observability of action occurrences as a function of the state in which they transpire. In this particular domain,

fluents of the form *attentive*(α) (read as “agent α is attentive”) are *perspective fluents*. Other fluents of this kind may be used for modeling group formation and collaborative action. These fluents may be manipulated directly by the agents via *perspective altering actions* such as *signal/distract* and *unite/disband*.

Assuming our domain signature is fixed, we represent the initial state of the domain via a collection of *initial state axioms* as follows:

initially $\mathcal{C}_{\{A,B,C\}}(\text{attentive}(\alpha) \wedge \text{bound}(A))$

initially $\mathcal{C}_{\{A,B,C\}}(\neg \mathcal{B}_A \text{allies}(A, C) \wedge \neg \mathcal{B}_B \text{allies}(B, C))$

initially $\mathcal{B}_C \text{allies}(A, C)$

Generally speaking, such axioms are statements of the form:

initially φ

where φ is a restricted kind of *modal formula* (Fagin et al. 1995) called a *belief formula* (Gelfond 2018; Baral et al. 2015), and have the informal reading of: “ φ is initially true.” The first axiom tells us that initially, it is a “commonly held belief” amongst agents A , B , and C that they are *attentive* and that agent A is *bound*.

When it comes to describing actions and their effects, in a multi-agent context it is important to note that in addition to direct effects, actions may have indirect effects based on whether or not they are observed by the agents. As a consequence, the *observability of actions* is also something we must describe within our action descriptions. With this in mind, the actions *signal* and *distract* in a straightforward manner, with their observability limited to those agents directly involved in the action occurrences, and attentive agents:

signal(α_1, α_2) **causes** *attentive*(α_2)

distract(α_1, α_2) **causes** $\neg \text{attentive}(\alpha_2)$

$\{\alpha_1, \alpha_2\}$ **observes** *signal*(α_1, α_2)

$\{\alpha\}$ **observes** *signal*(α_1, α_2) **if** *attentive*(α)

$\{\alpha_1, \alpha_2\}$ **observes** *distract*(α_1, α_2)

$\{\alpha\}$ **observes** *distract*(α_1, α_2) **if** *attentive*(α)

The first two axioms are *dynamic causal laws* in the fashion of the languages \mathcal{A} and \mathcal{AL} and describe the direct effects of the actions *signal* and *distract* respectively. The remainder are *perspective axioms* and define the rules by

which the agents of the domain observe individual action occurrences. The rule $[\{\alpha_1, \alpha_2\} \textbf{observes signal}(\alpha_1, \alpha_2)]$ states that the agents directly involved in the action occurrence are aware of it transpiring. Additionally, the rule $[\{\alpha\} \textbf{observes signal}(\alpha_1, \alpha_2) \textbf{if attentive}(\alpha)]$ extends this property to *attentive* agents.

In general, agents may *unite* in order to act together. In this particular domain, an agent must be $\neg\textit{bound}$ before he may unite with another agent to collaboratively perform some action. In addition, an agent will only unite with someone whom he believes is an ally. Once they are done collaborating, they may *disband*. This behavior is defined by the following axioms:

$$\begin{aligned} &\textit{unite}(\alpha_1, \alpha_2) \textbf{causes} \textit{united}(\alpha_1, \alpha_2) \\ &\textbf{executable} \textit{unite}(\alpha_1, \alpha_2) \textbf{if} \neg\textit{bound}(\alpha_1) \wedge \\ &\quad \neg\textit{bound}(\alpha_2) \wedge \mathcal{B}_{\alpha_1} \textit{allies}(\alpha_1, \alpha_2) \\ &\textit{disband}(\alpha_1, \alpha_2) \textbf{causes} \neg\textit{united}(\alpha_1, \alpha_2) \end{aligned}$$

The observation axioms governing the frames of reference of the agents with respect to occurrences of the actions *unite* and *disband* follow the same pattern as those for the actions *signal* and *distract* and hence are omitted from our presentation.

Now that we have finished defining the behaviors of the perspective altering actions, we turn our attention to the remaining actions. A single agent may *release* another agent causing him to no longer be *bound*. A pair of agents working together may *subdue* an agent, causing him to be *bound*.

$$\begin{aligned} &\textit{release}(\alpha_1, \alpha_2) \textbf{causes} \neg\textit{bound}(\alpha_2) \\ &\textit{subdue}(\alpha_1, \alpha_2, \alpha_3) \textbf{causes} \textit{bound}(\alpha_3) \\ &\textbf{executable} \textit{subdue}(\alpha_1, \alpha_2, \alpha_3) \textbf{if} \\ &\quad \textit{united}(\alpha_1, \alpha_2) \vee \textit{united}(\alpha_2, \alpha_1) \end{aligned}$$

The representation of the action *escape* is straightforward as well. Once an agent has escaped, he is *free*. Furthermore, we know that an agent, may only escape once his captor has been subdued (i.e. *bound*). The relevant observation axioms follow a now familiar pattern and are omitted due to space considerations.

$$\begin{aligned} &\textit{escape}(\alpha) \textbf{causes} \textit{free}(\alpha) \\ &\textbf{executable} \textit{escape}(\alpha_1) \textbf{if} \textit{captor}(\alpha_2, \alpha_1) \wedge \textit{bound}(\alpha_2) \wedge \\ &\quad (\neg\textit{united}(\alpha_1, \alpha_3) \vee \neg\textit{united}(\alpha_3, \alpha_1)) \end{aligned}$$

Lastly, an agent may *tell* another agent some fact about the domain. The action *tell* is a communication action, and is represented by the corresponding axiom:

$$\textit{tell}(\alpha_1, \alpha_2, \varphi) \textbf{communicates} \varphi$$

where φ is of the form $\textit{allies}(\alpha_1, \alpha_2)$. Agents may eavesdrop however, and therefore in the Escapee Domain, communication must be done with caution. For this domain, we assume that attentive agents are fully aware of what is said between their fellows. This assumption is encoded by the following observation axioms:

$$\begin{aligned} &\{\alpha_1, \alpha_2\} \textbf{observes} \textit{tell}(\alpha_1, \alpha_2, \varphi) \\ &\{\alpha_3\} \textbf{observes} \textit{tell}(\alpha_1, \alpha_2, \varphi) \textbf{if} \textit{attentive}(\alpha_3) \end{aligned}$$

The Transition Diagram

As is the case with other languages of this kind, the semantics of $m\mathcal{A}+$ is based on the notion of a transition diagram whose nodes represent *states of the domain* and whose edges are labeled by *actions*. In a multi-agent context such as the one presented here, states of the domain are represented by complex objects known as *Kripke worlds* (Fagin et al. 1995), and individual action occurrences are modeled by similar kinds of objects known as *update models* (van Benthem, van Eijck, and Kooi 2006). Each of these is a kind of graph itself whose nodes represent either *possible worlds* or *potential outcomes*, and whose edges represent what the agents of the domain believe about them. The transition function itself is defined by a function known as an *update execution*, which may be viewed as a graph product between a Kripke world representing a state of the domain and an update model describing a concrete action occurrence. The full definitions of these constructs are omitted from the presentation here, and we refer the reader to (Gelfond 2018; Baral et al. 2015) for a full treatment of the language $m\mathcal{A}+$. The aforementioned axioms constitute an *action description* of $m\mathcal{A}+$, which defines the domain's transition diagram, allowing us to precisely define and answer queries regarding *temporal projection*, *planning*, etc.

Some Concluding Thoughts

The use of *modal formulae* in $m\mathcal{A}+$ and consequent recasting of the notions of a state of the domain and requisite transition function as a graph product provide an interesting challenge for modern answer-set solvers such as *clingo*. Each object's definition on both a syntactic and semantic level render a direct translation of the language's semantics into a logic program computationally difficult due to the reliance of such systems on pre-grounding the logic program. It is the author's hope that the logic programming semantics of $m\mathcal{A}+$ in addition to being of general interest to the knowledge representation community, will also present a new set of challenge problems or benchmarks for subsequent developments in answer-set solvers.

References

- Baral, C.; Gelfond, G.; Pontelli, E.; and Son, T. C. 2015. Multi-Agent Action Modeling through Action Sequences and Perspective Fluents. In *Proceedings of the AAAI Spring Symposium on Common Sense Reasoning 2015*, AAAI '15.
- Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning About Knowledge*. MIT Press.
- Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2012. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers.
- Gelfond, G. 2018. *Representing and Reasoning about Dynamic Multi-Agent Domains: An Action Language Approach*. Ph.D. Dissertation, Arizona State University.
- van Benthem, J.; van Eijck, J.; and Kooi, B. 2006. Logics of communication and change. *Information and Computation* 204(11):1620–1662.