Bounded Practical Social Reasoning in the ESB Framework

Iain Wallace iain.wallace@ed.ac.uk

Michael Rovatsos michael.rovatsos@ed.ac.uk

ABSTRACT

Each agent in a multi-agent system will have its own goals and methods of reasoning about what actions to take. For an individual agent, reasoning about others is often a complex process. The Expectation-Strategy-Behaviour (ESB) framework provides an intuitive graph-based method to represent and simplify reasoning.

This poster introduces ESB through a simple example based on the card game Rummy, then presents one way in which the ESB design allows for simpler agent reasoning, and a simpler implementation.

THE EXAMPLE

Cards provide a good domain, as an opponent's strategies are known, but defined by the unknown cards they hold - an intuitive example of reasoning about another agent. Here we show a model of an opponent's reasoning during a game of Rummy, specifically that surrounding the pickup of a 2♢. The other agent is assumed to be collecting a run (e.g. 2♢,3♢,4♢) or a set (e.g. 2♢, 2♡, 2♣).

EXPECTATIONS

An **expectation (E)** is a **conditional (C) belief (Φ)** regarding a statement whose truth status will be eventually **verified** by a **test (T)** and **reacted (ρ)** on by the agent who holds it. **Table 1** below shows the expectations for this example. Tests are split into + and - events that confirm or disappoint the expectation. **Responses (ρ+,ρ-)** add or remove expectations.

Consider agent A expecting agent B to execute some action upon promising to do so. The expected belief is "B will perform an action (Φ) if he has promised to do so (C)". The test (T) is observing the action or its consequences. The response might be A rewarding B if the promise is kept (ρ+), or not (ρ-); or, the expectation itself might be modified, for example altering the level of trust towards B.

STRATEGIES

The graph on the right is the **expectation graph**. The vertices are states containing active expectations, and the edges mark the transitions between states **specified by the responses (ρ)**. A **strategy** is a restriction of the graph based on current state. The highlighted portion shown is a **strategy applied to state 5** considering states to depth 1.

BEHAVIOURS

1. Only discard a 2 if *B* is not collecting 2s.
2. Only discard a 3 if *B* is not collecting 3s.
3. Only discard cards I don't expect *B* to be collecting in the future.

Behaviours are "**actions**" linked to **conditions** on the **strategy graph**. Therefore, conditions are on current expectations and possible future expectations - those in states reachable from the current state according to the strategy graph. **For example, in state 5, E^2_Φ is a possible future expectation.**

The above behaviour rules are fairly general - in particular the meaning of rule 3 is unclear. These rules can be translated into **explicit behaviour rules** conditioned on the expectations used in this example.

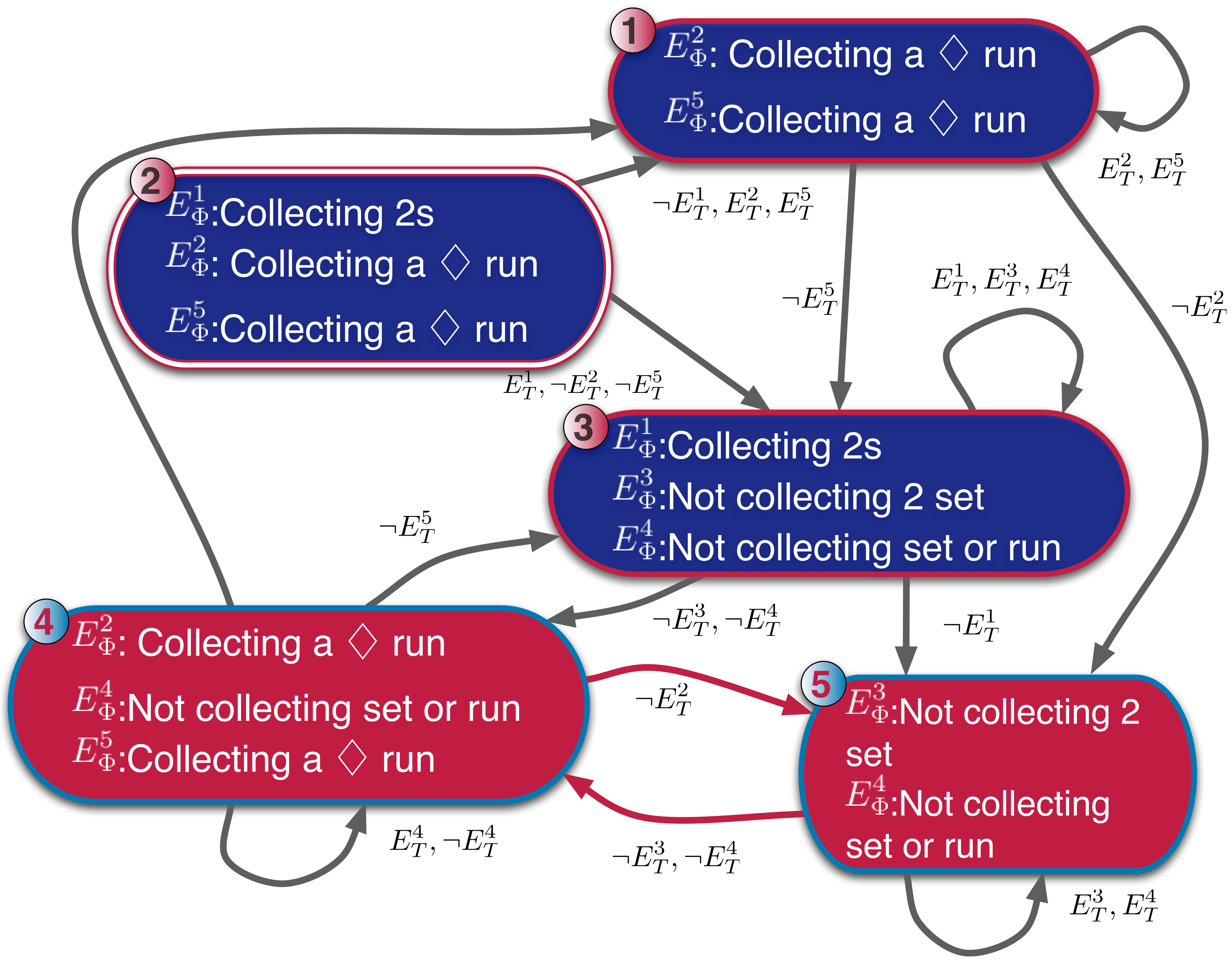
Table 1: The agent's expectations. Initially it holds the set {1,2,5}.

N	C	Φ	T	ρ^+	ρ^-
1	B picked up 2♢	B is collecting 2s	$+$(B picks up a 2) $-$(B ignores/discards a 2)	remove({2,5}) add({3,4})	remove({1})
2	B picked up 2♢	B is collecting ♢ run	$+$(B picks up member of 2♢ run) $-$(B ignores/discards member of 2♢ run)	remove({1,3,4}) add({5})	remove({2,5}) add({3,4})
3	B discarded 2♣	B not collecting 2s	$+$(B ignores a 2) $-$(B picks up a 2)	-	remove({1,3}) add({2,5})
4	B ignored a 2	B not after 2s for run or set	$+$(B ignores a 2) $-$(B picks up a 2)	-	remove({1,3}) add({2,5})
5	B picked up 3♢	B is collecting ♢ run	$+$(B picks up another member of 3♢ run) $-$(B ignores/discards member of 3♢ run)	remove({1,3,4}) add({2})	remove({2,5}) add({1,3,4})

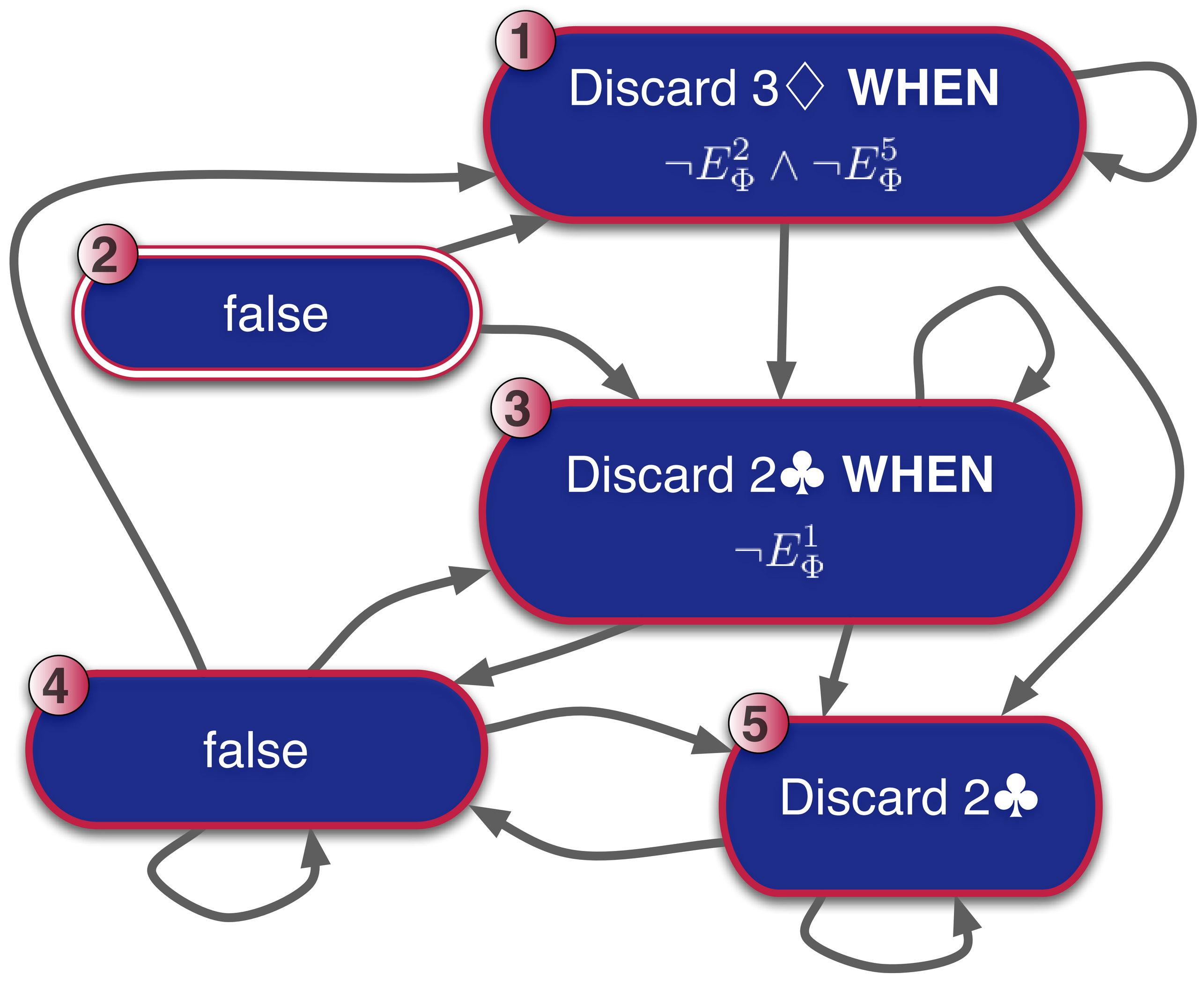
Explicit Behaviour Rules

1. (a) Discard 2♢ if $\neg E^1_\Phi \wedge \neg E^2_\Phi \wedge \neg E^5_\Phi$
   (b) Discard 2♣ if $\neg E^1_\Phi$
2. Discard 3♢ if $\neg E^2_\Phi \wedge \neg E^5_\Phi$
3. (a) Discard 2♢ if in all accessible states $\neg E^1_\Phi \wedge \neg E^2_\Phi \wedge \neg E^5_\Phi$
   (b) Discard 2♣ if in all accessible states $\neg E^1_\Phi$
   (c) Discard 3♢ if in all accessible states $\neg E^2_\Phi \wedge \neg E^5_\Phi$

An Expectation Graph



SIMPLIFYING BEHAVIOURS



Behaviour conditions such as "in all accessible states" or "in some accessible states" could be expensive to check in large graphs. The application of a strategy to create the **strategy graph** reduces the problem, but does not eliminate it. One generalised technique that can be applied to bound agent reasoning (and thus simpify implementation) is to **re-write behaviour rules** to include state. The intuitive reasoning behind the algorithm used is that considering each behaviour in each state in turn, there are three possibilities:

1. The condition is always true.
2. The condition is always false.
3. In this state, some minimum combination of expectations decides the condition.

Pre-applying the behaviours with these rules gives **the simplified graph** above, where each state shows the behaviour rules that can apply.

School of informatics