Ninth Workshop on Automated Reasoning
Bridging the Gap between Theory and Practice

Collected Abstracts

Toby Walsh (editor)
Cork Constraint Computation Center
University College Cork
Ireland

Towards automated generation of beliefs in BDI agents

A. Basukoski, A. Bolotov
Harrow School of Computer Science, University of Westminster,
Harrow, Watford Road, HA1 3TP.
E-mail: {A. Basukoski,A.Bolotov}@westminster.ac.uk

Abstract

Formal BDI frameworks such as LORA [5] which use temporal logic as a tool for describing beliefs about the environment consider an agent behavior in the following control loop:

```
Algorithm: Agent Control Loop
1. while true
2.     observe the world;
3.     update internal world model;
4.     deliberate about what
           intention to achieve next;
5.     use means ends reasoning to
            get a plan for the
             next intention;
6.     execute the plan;
7. end while.
```

Whereas the vast majority of agent literature focuses on steps 4, 5, and 6, we are interested in the transition 2→3 which forms a link between the perceived world and the way knowledge about that world is represented within the BDI framework. One of the efficient ways of specifying beliefs is to use combinations of temporal or dynamic logics and modal logics (see, for example, [1, 3]). We define a formal structure from which Beliefs can be automatically extracted in a way suitable for such formal representation.

Thus, we introduce an algorithm to incrementally generate this structure, which we call a 'A Temporal Lattice'. We invoke the standard technique of the Formal Concept Analysis [4, 2], where an observation, ρ, is formally represented as $\rho \subseteq E \times I$, where E is a set of *extents*, and I is a set of *intents* [2]. Since ρ can be viewed as an expression in propositional logic [2], the set of observations forms the alphabet, $\mathcal{A}$ for labelling the nodes in a graph being constructed.

At each step, we assume that the deliberation on which action (from the set of possible actions) must be next taken is done by a deliberation function which forms part of the BDI architecture. Thus, viewing this deliberation function as a transition function, we unwind the structure as a graph such that, given a state, n, we make a non-deterministic choice of the successor state, m.

This forms a sequences of states $\tau_0, \tau_1, \tau_2, \ldots$, which are linked by actions $act \in Act$. Additionally, extending the alphabet $\mathcal{A}$ by **true** and **false** (with their standard meaning accepted in classical logic) we label them by the corresponding expressions from $\mathcal{A}$ (the initial state is labelled by **false**). We refer to this sequence as level 0 ($n = 0$) of the Temporal Lattice to be constructed at the next stage.

Given a sequence τ, (see Figure 1) the initial node of the Temporal Lattice, is the initial state of τ, and we denote this node as $\omega_{0,0}$, indicating that it occurs at the 0-th point of time and 0-th level. The successor node, $\omega_{1,0}$ is the state τ_1 of τ and has the same label as the state τ_1, say, ρ_k. Now, we build the node $\omega_{1,1}$ such that its label, ρ_l, satisfies the following condition:

$$\rho_l \equiv (\mathbf{false} \vee \rho_k) \equiv \rho_k$$

and the node $\omega_{1,-1}$ such that its label, ρ_m, satisfies the following condition:

$$\rho_m \equiv (\mathbf{false} \wedge \rho_k) \equiv \mathbf{false}.$$

The second index of the nodes $\omega_{1,1}$ and $\omega_{1,-1}$ indicate their level in the lattice, i.e. level 1 and level -1,

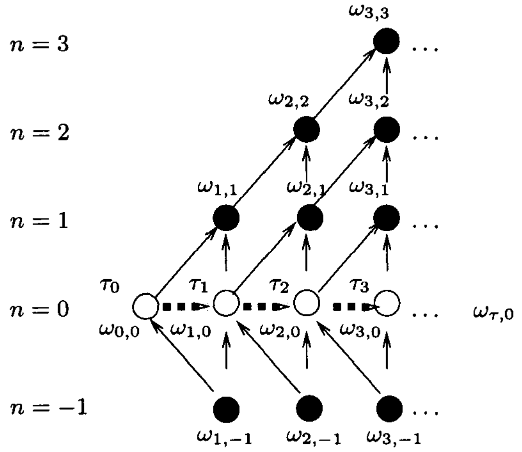

Figure 1: Temporal Lattice

respectively. Repeating this procedure again on subsequent nodes, we derive the Temporal Lattice (the fact the structure is indeed a lattice follows from the construction algorithm).

The analysis of the Lattice enables us to

- generate abstractions,
- extract expressions (in linear-time temporal logic) for the deliberation function,
- derive models of linear computations where these formulae are satisfied.

We believe that the Temporal Lattice is an efficient method (at every time point, i, we generate at most $2 \times i$ nodes in the lattice) to generate Knowledge and Beliefs incrementally from raw percepts and is therefore useful in agent applications where

1. there is limited perception of the environment
2. there is no prior knowledge about the environment
3. and the environment is dynamic and nondeterministic.

References

[1] C. Dixon, M. Fisher, and A. Bolotov. Clausal Resolution in a Logic of Rational Agency. *Artificial Intelligence*, 2002. To appear.

[2] B. Gantner and R. Wille. *Formal Concept Analysis.* Springer, 1996.

[3] U. Hustadt, C. Dixon, R. Schmidt, M. Fisher, J. Meyer, and W. van der Hoek. Reasoning about agents in the karo framework. In *Proceedings of the Eighth International Symposium on Temporal Representation and Reasoning (TIME-01)*, pages 206–213, Cividale del Friuli, Italy, June 2001. Computer Society Press.

[4] J.F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Brooks Cole Publishing Co, 1999.

[5] M. Wooldridge. *Reasoning about Rational Agents.* MIT Press, July 2000.