An Agent Logic Tableau System

Renate A. Schmidt and Dmitry Tishkovsky

Department of Computer Science, University of Manchester, {schmidt,dmitry}@cs.man.ac.uk

Introduction. Multi-agent systems are important to many areas of computer science, starting from specifying and verifying multi-threaded systems and multiprocessor computers to modelling populations of robots which have animal or human-like behaviour. The behaviour of agents is typically very complicated and agent systems are therefore complex. In previous work we have investigated a family of logic-based multi-agent systems which are based on an agent logic called *BDL* [3]. *BDL* is a logic for modelling and reasoning about the dynamic activities and informational attitudes of agents. It is a *PDL*-like logic enhanced with modal operators for modelling the belief and knowledge states of agents. In contrast to other *PDL*-like agent logics, *BDL* provides a formalism for accommodating certain inter-agent dependencies, and for integrating doxastic aspects with the dynamic activities of the agents. In *BDL* interaction between agents is possible via shared actions. There are two kinds of actions: abstract actions and concrete actions. Abstract actions are virtual actions which can be performed by any agent, whereas concrete actions are the actions that a particular agent will actually perform. In *BDL* the test operator of *PDL* is replaced by an informational test operator which confirms the agents' beliefs about some property of the agent system rather than confirming the absolute truth of the property, which is what the *PDL* test operator does. The common belief operator can be expressed in *BDL* which makes it more expressive than *PDL*. However, despite its expressiveness, *BDL* is a decidable logic. Unfortunately, neither the Hilbert axiomatisation nor the procedure implicit in the small model theorem provide practical decision procedures for the reasoning problem of *BDL*. This paper is a first step toward developing more practical automated decision procedures for *BDL*. In particular, we present a tableau calculus for *BDL*. The calculus is based on the prefix tableau approach developed for *(converse) PDL* and *K45* in [1, 2].

Language. The language $\mathcal{L}$ of the logic *BDL* [3] is defined over a set $\mathrm{Var} = \{p, q, r, \ldots\}$ of propositional variables, a set $\mathrm{AtAA} = \{a, b, c, \ldots\}$ of *abstract atomic action* variables, and a set $\mathrm{Ag} = \{i, j, k, \ldots\}$ of agent variables. The logical connectives include $\wedge$ and $\neg$, the *informational test* operator ⁇, the standard *PDL* connectives on action formulae, $\cup$, ;, and $*$, the dynamic modal operator $[\_]$ (box) and a family of modal box operators $\mathbf{B}_i$ for all $i \in \mathrm{Ag}$. Abstract actions, concrete actions and formulae are defined by simultaneous induction so that AA, CA and For are the smallest sets satisfying the following conditions.

- $\mathrm{AtAA} \subseteq \mathrm{AA}$, $\mathrm{Var} \subseteq \mathrm{For}$.
- $\phi⁇, \alpha^*, \alpha \cup \beta, \alpha;\beta \in \mathrm{AA}$, if $\phi \in \mathrm{For}$ and $\alpha, \beta \in \mathrm{AA}$.
- $\alpha_i \in \mathrm{CA}$, if $\alpha \in \mathrm{AA}$ and $i \in \mathrm{Ag}$.
- $\gamma^*, \gamma \cup \delta, \gamma;\delta \in \mathrm{CA}$, if $\gamma, \delta \in \mathrm{CA}$.
- $\phi \wedge \psi, \neg\phi, [\gamma]\phi, \mathbf{B}_i\phi \in \mathrm{For}$, if $\phi, \psi \in \mathrm{For}$ and $\gamma \in \mathrm{CA}$.

As usual the following are assumed to be defined formula connectives: $\rightarrow$, $\perp$, $\top$, $\vee$, $\leftrightarrow$, and $\langle\_\rangle$ is the dual of $[\_]$. The formula $\mathbf{B}_i p$ is read as 'agent i believes p'. We reserve the symbols α and β for abstract actions, and the symbols γ and δ for concrete actions.

Let the set of *prime actions* be

$$\mathrm{PrCA} =^{\mathrm{def}} \{a_i \mid a \in \mathrm{AtAA}, i \in \mathrm{Ag}\} \cup \{(\top⁇)_i \mid i \in \mathrm{Ag}\}.$$

Semantics. A *BDL model* is a tuple $M = \langle S, Q, R, \models\rangle$ of a non-empty set S of states, a mapping Q of concrete actions γ to binary relations $Q(\gamma)$, a mapping R from the set of agents into the set of transitive and Euclidean relations over S, and $\models$ is the truth relation. Any *BDL* model must satisfy the following conditions.

(1) $Q(\gamma \cup \delta) = Q(\gamma) \cup Q(\delta)$
(2) $Q(\gamma;\delta) = Q(\gamma) \circ Q(\delta)$
(3) $Q((\gamma)^*) = (Q(\gamma))^*$
(4) $Q((\alpha \cup \beta)_i) = Q(\alpha_i \cup \beta_i)$
(5) $Q((\alpha;\beta)_i) = Q(\alpha_i;\beta_i)$
(6) $Q((\alpha^*)_i) = Q((\alpha_i)^*)$

where $\circ$ denotes relational composition and $*$ is the reflexive transitive closure operator. In the following conditions s denotes any state in S.

(7) $M, s \models \neg\phi \iff M, s \not\models \phi$
(8) $M, s \models \phi \wedge \psi \iff M, s \models \phi \& M, s \models \psi$
(9) $M, s \models [\gamma]\phi \iff \forall t\,((s, t) \in Q(\gamma) \Rightarrow M, t \models \phi)$
(10) $M, s \models \mathbf{B}_i\phi \iff \forall t\,((s, t) \in R_i \Rightarrow M, t \models \phi)$

The semantics of the informational test operator is given by:

(11) $Q((\phi⁇)_i) = \{(s, t) \in R_i \mid M, t \models \mathbf{B}_i\phi\}$.

(In this paper, we write R_i instead of $R(i)$.) A formula ϕ is said to be true in a state s of a model M if $M, s \models \phi$, and ϕ is *satisfiable* if there exists a model M and a state in M where ϕ is true.

The tableau calculus. In [1, 2] prefix tableau algorithms are presented for solving the satisfiability problem in *PDL*, *converse PDL* and *K45*. In this paper we define a prefix tableau calculus for the satisfiability problem in the agent logic *BDL*.

A *prefix* is an alternating sequence of integers and prime actions. For convenience we will use symbols R instead of $\top⁇$ in the prefixes, i.e. the definition of a prefix is:

$$\sigma =^{\mathrm{def}} 0 \mid \sigma.a_i.n \mid \sigma.R_i.n$$

A *prefixed formula* is a pair $\sigma : \phi$ of a prefix σ and a formula ϕ. A *tableau* is a rooted tree in which nodes are labelled by prefixed formulae. A *branch* $\mathcal{B}$ in a tableau is a path from the root to a leaf.

Figure 1 lists the rules of the prefix tableau calculus for *BDL*. The rules for formulae involving concrete actions are the same as for *test-free PDL* in [1]. To handle the $*$ operator we use introduced symbols $X^{\sigma}_{\neg[\gamma^*]\phi}$ (called *eventualities*) which are indexed by prefixes and formulae to remember where the symbols were introduced. Two eventualities $X^{\sigma}_{\neg[\gamma^*]\phi}$ and $X^{\tau}_{\neg[\delta^*]\psi}$ are regarded to be *equal* if $[\gamma^*]\phi = [\delta^*]\psi$. The rules for abstract action formulae and informational test are rewrite rules based on validities in *BDL*. Finally, there are the familiar *K45* rules for $\mathbf{B}_i$ [2].

Local rules:

$$(\bot)\ \frac{\sigma : \psi,\ \ \sigma : \neg\psi}{\sigma : \bot} \qquad (\neg)\ \frac{\sigma : \neg\neg\psi}{\sigma : \psi}$$

$$(\wedge)\ \frac{\sigma : \psi \wedge \phi}{\sigma : \psi,\ \ \sigma : \phi} \qquad (\neg\wedge)\ \frac{\sigma : \neg(\psi \wedge \phi)}{\sigma : \neg\psi \ \mid\ \sigma : \neg\phi}$$

$$(\neg\cup^a)\ \frac{\sigma : [(\alpha \cup \beta)_i]\psi}{\sigma : [\alpha_i \cup \beta_i]\psi} \qquad (\cup^a)\ \frac{\sigma : \neg[(\alpha \cup \beta)_i]\psi}{\sigma : \neg[\alpha_i \cup \beta_i]\psi}$$

$$(\neg\,;^a)\ \frac{\sigma : [(\alpha\,;\beta)_i]\psi}{\sigma : [\alpha_i\,;\beta_i]\psi} \qquad (;^a)\ \frac{\sigma : \neg[\alpha\,;\beta]\psi}{\sigma : \neg[\alpha_i\,;\beta_i]\psi}$$

$$(\neg^{*a})\ \frac{\sigma : [(\alpha^*)_i]\psi}{\sigma : [(\alpha_i)^*]\psi} \qquad (^{*a})\ \frac{\sigma : \neg[(\alpha^*)_i]\psi}{\sigma : \neg[(\alpha_i)^*]\psi}$$

$$(\neg\cup)\ \frac{\sigma : [\gamma \cup \delta]\psi}{\sigma : [\gamma]\psi,\ \ \sigma : [\delta]\psi} \qquad (\cup)\ \frac{\sigma : \neg[\gamma \cup \delta]\psi}{\sigma : \neg[\gamma]\ \mid\ \sigma : \neg[\delta]\psi}$$

$$(\neg\,;)\ \frac{\sigma : [\gamma\,;\delta]\psi}{\sigma : [\gamma][\delta]\psi} \qquad (;)\ \frac{\sigma : \neg[\gamma\,;\delta]\psi}{\sigma : \neg[\gamma][\delta]\psi}$$

$$(\neg^*)\ \frac{\sigma : [\gamma^*]\psi}{\sigma : \psi,\ \ \sigma : [\gamma][\gamma^*]\psi} \qquad (^*)\ \frac{\sigma : \neg[\gamma^*]\psi}{\sigma : X^{\sigma}_{\neg[\gamma^*]\psi}}$$

$$(X)\ \frac{\sigma : X^{\tau}_{\neg[\gamma^*]\psi}}{\sigma : \neg\psi\ \mid\ \sigma : \psi,\ \ \sigma : \neg[\gamma]\neg X^{\tau}_{\neg[\gamma^*]\psi}}$$

$$(\neg?)\ \frac{\sigma : [(\psi?)_i]\phi}{\sigma : \mathbf{B}_i(\mathbf{B}_i\psi \to \phi)} \qquad (?)\ \frac{\sigma : \neg[(\psi?)_i]\phi}{\sigma : \neg\mathbf{B}_i(\mathbf{B}_i\psi \to \phi)}$$

Transition rules:

$$(\neg[a_i])\ \frac{\sigma : \neg[a_i]\psi}{\sigma.a_i.n : \neg\psi}\ \text{ where } n \text{ is new to the branch}$$

$$([a_i])\ \frac{\sigma : [a_i]\psi,\ \ \sigma.a_i.n : \theta}{\sigma.a_i.n : \psi}$$

$$(\neg\mathbf{B}_i)\ \frac{\sigma : \neg\mathbf{B}_i\psi}{\sigma.R_i.n : \neg\psi}\ \text{ where } n \text{ is new to the branch}$$

$$(\mathbf{B}_i)\ \frac{\sigma : \mathbf{B}_i\psi,\ \ \sigma.R_i.n : \theta}{\sigma.R_i.n : \psi}$$

Theory rules:

$$(4)\ \frac{\sigma : \mathbf{B}_i\psi,\ \ \sigma.R_i.n : \theta}{\sigma.R_i.n : \mathbf{B}_i\psi} \qquad (5)\ \frac{\sigma.R_i.n : \mathbf{B}_i\psi}{\sigma : \mathbf{B}_i\psi}$$

Fig. 1. Prefix tableau expansion rules for *BDL*.

A prefix is *reduced* if all the rules except the rules $(\neg[a_i])$ and $(\neg\mathbf{B}_i)$ have been applied to it and it is *fully reduced* if all rules have been applied to it. A prefix σ is a *copy* of τ in a branch $\mathcal{B}$ if the set of all formulae prefixed by σ in $\mathcal{B}$ is equal to the set of formulae prefixed by τ in $\mathcal{B}$. A reduced prefix σ is *collapsed* into τ in $\mathcal{B}$ iff τ is a fully reduced copy and a proper sub-prefix of σ.

A branch $\mathcal{B}$ is *completed* iff all the prefixes in $\mathcal{B}$ are fully reduced or collapsed into some prefixes in $\mathcal{B}$. An eventuality $X^{\sigma}_{\neg[\gamma^*]\phi}$ is *collapsed* into an (equal) eventuality $X^{\tau}_{\neg[\gamma^*]\phi}$ iff there are prefixes μ and ν such that μ is collapsed into ν and both $\mu : X^{\sigma}_{\neg[\gamma^*]\phi}$ and $\nu : X^{\tau}_{\neg[\gamma^*]\phi}$ are in $\mathcal{B}$. $X^{\sigma}_{\neg[\gamma^*]\phi}$ is *fulfilled* in a branch $\mathcal{B}$ iff either there is a prefix τ such that both $\tau : X^{\sigma}_{\neg[\gamma^*]\phi}$ and $\tau : \neg\phi$ are in $\mathcal{B}$, or it is collapsed into some other fulfilled eventuality in $\mathcal{B}$. A branch is *contradictory* if it contains $\sigma : \bot$ for some σ. A π-completed branch is *ignorable* iff it contains an unfulfilled eventuality. A tableau is *closed* if all its branches are either contradictory or ignorable. A tableau is *open* if it is not closed. We say that ϕ *has an open (closed) tableau* if the tableau started from $0 : \phi$ is open (closed).

Informally, the algorithm works as follows. We apply all the rules which do not generate new prefixes to prefixed formulae of a given branch until we find a contradiction or no more of these rule applications are possible. In the first case the branch is closed and can therefore be abandoned so that the derivation can be continued on another branch. In the second case we collect all the prefixes on the branch which are not fully reduced and check whether they collapse into some fully reduced prefixes on the branch. If there is no fully reduced copy of such a collected prefix then we apply the rest of the rules to it to generate new prefixes and proceed with these generated prefixes. Otherwise, i.e. if there is nothing is left to do on the branch, we test the branch for unfulfilled eventualities, thereby attempting to find an unsuccessful loop and, then, to close and abandon the branch. By identifying prefixes, equivalent with respect to the properties they satisfy, infinite open branches can be avoided and termination therefore guaranteed.

In the tableau calculus the complex conditions on substitution which are part of the axiomatic definition of *BDL* [3] are avoided, by using different symbols for terms from different sorts. It is worth mentioning though, that any application of a tableau rule to a prefixed formula implicitly uses a unification algorithm which satisfies the substitution conditions for *BDL*.

Theorem 1 (Soundness and completeness). *Any formula is satisfiable in BDL iff it has an open BDL-tableau.*

Theorem 2 (Termination). *For any BDL formula ϕ the tableau constructed for $0 : \phi$ is finite.*

References

1. G. de Giacomo and F. Massacci. Combining deduction and model checking into tableaux and algorithms for converse-PDL. *Inform. and Computat.*, 162:117–137, 2000.
2. F. Massacci. Single step tableaux for modal logics. *J. of Automated Reasoning*, 24:319–364, 2000.
3. R. A. Schmidt and D. Tishkovsky. Multi-Agent Logic of Dynamic Belief and Knowledge. In *Proc. JELIA'2002*, vol. 2424 of *LNAI*, pp. 38–49. Springer, 2002.