Using Strategic Logics to Reason about Agent Programs*

Nitin Yadav
RMIT University
Melbourne, Australia
nitin.yadav@rmit.edu.au

Sebastian Sardina
RMIT University
Melbourne, Australia
sebastian.sardina@rmit.edu.au

Abstract

We propose a variant of Alternating-time Temporal Logic (ATL) grounded in the agents' operational know-how, as defined by their libraries of abstract plans. In our logic, it is possible to refer to "rational" strategies for agents developed under the Belief-Desire-Intention agent paradigm. This allows us to express and verify properties of BDI systems using ATL-type logical frameworks.

1 Introduction

In this paper we report on our work on using strategic logics for reasoning about the ability of agents developed under the Belief-Desire-Intention (BDI) agent-oriented programming tradition [Bratman *et al.*, 1988; Rao and Georgeff, 1992; Bordini *et al.*, 2006], a popular paradigm for building multi-agent systems. In particular, we aim at providing a verification framework that will allow us to reason about what a group of agents' can achieve with their available know-how capabilities under the BDI paradigm, which enables abstract plans to be combined and used in real-time under the principles of practical reasoning [Bratman *et al.*, 1988].

While recent work (e.g., [Alechina *et al.*, 2007; 2008; Dastani and Jamroga, 2010]) has tried to better bridge the gap between formal logic and practical programming, this was done by restricting the logic's models to those that satisfy the transition relations of agents' plans and without reasoning about what a group of agents could achieve *if they had* specific capabilities.

To achieve our objectives, we adapt ATLES, a version of ATL (Alternating-time Temporal Logic) [Alur *et al.*, 2002] with Explicit Strategies [Walther *et al.*, 2007]. In ATL, one is interested in checking formulas of the form $\langle\langle A\rangle\rangle\varphi$ expressing that the coalition team of agents A has a joint strategy for guaranteeing that the temporal property φ holds. ATLES [Walther *et al.*, 2007] extends ATL to accommodate formulas of the form $\langle\langle A\rangle\rangle_\rho\varphi$ denoting that coalition A has a joint strategy for ensuring φ, *when some agents are committed to* specific *strategies, as specified by so-called commitment function* ρ. Here we go further and develop a logic—called BDI-ATLES—in which commitments can be tied directly to the know-how of agents under the notion of practical reasoning embodied by the BDI paradigm [Bratman *et al.*, 1988; Rao and Georgeff, 1992]: *the only strategies that can be employed by a BDI agent are those that ensue by the (rational) execution of its predefined plans, given its goals and beliefs*.[1] To achieve this, our framework includes a construct $\langle\langle A\rangle\rangle_{\omega,\varrho}\varphi$ stating that coalition A has a joint strategy for ensuring φ, *under the assumptions that some agents in the system are BDI-style agents* with capabilities and goals as specified by assignments ω and ϱ, respectively. For instance, a formula like $\langle\langle A\rangle\rangle_{\emptyset,\emptyset}\varphi \supset \langle\langle A\rangle\rangle_{\omega,\varrho}\varphi$ can be used to check if coalition A has enough know-how and motivations to carry out a task φ that is indeed physically feasible for the coalition.

2 Preliminaries

2.1 ATL/ATLES Logics of Coalitions

Alternating-time Temporal Logic (ATL) [Alur *et al.*, 2002] is a logic for reasoning about the ability of agent coalitions in *multi-agent game structures*. ATL formulae are built by combining propositional formulas, the usual temporal operators—namely, $\bigcirc$ ("in the next state"), $\Box$ ("always"), $\Diamond$ ("eventually"), and $\mathcal{U}$ ("strict until")—and a *coalition path quantifier* $\langle\langle A\rangle\rangle$ taking a set of agents A as parameter. As in CTL, which ATL extends, temporal operators and path quantifiers are required to alternate. Intuitively, an ATL formula $\langle\langle A\rangle\rangle\phi$, where A is a set of agents, holds in an ATL structure if by suitably choosing their moves, the agents in A *can force* ϕ *true*, no matter how other agents happen to move. The semantics of ATL is defined in so-called *concurrent game structures* where, at each point, all agents simultaneously choose their moves from a finite set, and the next state deterministically depends on such choices. More concretely, an ATL structure is a tuple $\mathcal{M} = \langle \mathcal{A}, Q, \mathcal{P}, Act, d, \mathcal{V}, \sigma\rangle$, where $\mathcal{A} = \{1, \ldots, k\}$ is a finite set of agents, Q is the finite set of states, $\mathcal{P}$ is the finite set of propositions, *Act* is the set of all

*The paper on which this extended abstract is based was the recipient of the best paper award of the 2012 European Conference on Logics in Artificial Intelligence [Yadav and Sardina, 2012]
We acknowledge the support of the Australian Research Council (DP1094627 & DP120100332).

[1] The notion of "rationality" used here is that found in the literature on BDI agent programming (as reasonable constraints on how the various mental modalities may interact), rather than that common in game-theory (generally captured via *solution concepts*).

domain actions, $d : \mathcal{A} \times Q \mapsto 2^{Act}$ indicates all available actions for an agent in a state, $\mathcal{V} : Q \mapsto 2^{\mathcal{P}}$ is the valuation function stating what is true in each state, and $\sigma : Q \times Act^{|\mathcal{A}|} \mapsto Q$ is the transition function mapping a state q and a joint-move $\vec{a} \in \mathcal{D}(q)$—where $\mathcal{D}(q) = \times_{i=1}^{|\mathcal{A}|} d(i, q)$ is the set of legal joint-moves in q —to the resulting next state q'.

To provide semantics to formulas $\langle\langle\cdot\rangle\rangle\varphi$, ATL relies on the notion of agent strategies. Technically, an ATL *strategy* for an agent *agt* is a function $f_{agt} : Q^+ \mapsto Act$, where $f_{agt}(\lambda q) \in d(agt, q)$ for all $\lambda q \in Q^+$, stating a particular action choice of agent *agt* at path λq. A *collective strategy* for group of agents $A \subseteq \mathcal{A}$ is a set of strategies $F_A = \{f_{agt} \mid agt \in \mathcal{A}\}$ providing one specific strategy for each agent $agt \in A$. For a collective strategy F_A and an initial state q, it is not difficult to define the set $out(q, F_A)$ of all *possible outcomes* of F_A starting at state q as the set of all computation paths that may ensue when the agents in A behave as prescribed by F_A, and the remaining agents follow any arbitrary strategy [Alur *et al.*, 2002; Walther *et al.*, 2007]. The semantics for the coalition modality is then defined as follows (here ϕ is a *path formula*, that is, it is preceded by $\bigcirc$, $\Box$, or $\mathcal{U}$, and $\mathcal{M}, \lambda \models \phi$ is defined in the usual way [Alur *et al.*, 2002]):

$\mathcal{M}, q \models \langle\langle A\rangle\rangle\phi$ *iff* there is a collective strategy F_A such that for all computations $\lambda \in out(q, F_A)$, we have $\mathcal{M}, \lambda \models \phi$.

The coalition modality only allows for implicit (existential) quantification over strategies. In some contexts, though, it is important to refer to strategies explicitly in the language, e.g., can a player win the game if the opponent plays a specified strategy? To address this limitation, Walther *et al.* [Walther *et al.*, 2007] proposed ATLES, an extension of ATL where the coalition modality is extended to $\langle\langle A\rangle\rangle_\rho$, where ρ is a *commitment function*, that is, a partial function mapping agents to so-called *strategy terms*. Formula $\langle\langle A\rangle\rangle_\rho\phi$ thus means that *"while the agents in the domain of ρ act according to their commitments, the coalition A can cooperate to ensure ϕ as an outcome."*

The motivation for our work stems from the fact that ATLES is agnostic on the source of the strategic terms: all meaningful strategies have already been identified. In the context of multi-agent systems, it may not be an easy task to identify those strategies compatible with the agents' behaviors, as those systems are generally built using programming frameworks [Bordini *et al.*, 2006] that are very different from ATL(ES).

2.2 BDI Programming

The BDI agent-oriented programming paradigm is a popular and successful approach for building agent systems, with roots in philosophical work on rational action [Bratman *et al.*, 1988] and a plethora of programming systems available, such as JACK, JASON, JADEX, 2APL [Bordini *et al.*, 2006], and GOAL [de Boer *et al.*, 2007], among others.

A typical BDI agent continually tries to achieve its goals (or desires) by selecting an adequate plan from its *plan library* given its current beliefs, and placing it into the *intention base* for execution. The agent's plan library Π encodes the standard operational knowledge of the domain by means of a set of *plan-rules* (or "recipes") of the form $\phi[\alpha]\psi$: *plan α is a reasonable plan to adopt for achieving ψ when (context) condition ϕ is believed true*. For example, walking towards location x from y is a reasonable strategy, if there is a short distance between x and y (and the agent wants to be eventually at location x). Conditions ϕ and ψ are (propositional) formulas talking about the current and goal states, respectively. Though different BDI languages offer different constructs for crafting plans, most allow for sequences of domain actions that are meant to be directly executed in the world (e.g., lifting an aircraft's flaps), and the posting of (intermediate) *sub-goals* $!\varphi$ (e.g., obtain landing permission) to be resolved. The intention base, in turn, contains the current, partially executed, plans that the agent has already *committed to* for achieving certain goals. Current intentions being executed provide a screen of admissibility for attention focus [Bratman *et al.*, 1988].

Though we do not present it here for lack of space, most BDI-style programming languages come with a clear single-step semantics basically realizing [Rao and Georgeff, 1992]'s execution model in which *(rational) behavior arises due to the execution of plans from the agent's plan library so as to achieve certain goals relative to the agent's beliefs*.

3 BDI-ATLES: ATL for BDI Agents

The overarching aim of this section is to allow the encoding of BDI applications in ATL in a principled manner. In particular, we imagine a BDI developer interested in what agents can achieve given their goals and capabilities. Hence, we envision BDI agents defined with a set of *goals* and so-called *capabilities* [Busetta *et al.*, 1999; Padgham and Lambrix, 2005]. Generally speaking, a capability is a set/module of procedural knowledge (i.e., plans) for some functional requirement. An agent may have, for instance, the Navigate capability encoding all plans for navigating an environment. Equipped with a set of capabilities, a BDI agent executes actions as per plans available so as to achieve her goals, e.g., exploring the environment.

In this work, we shall consider plans consisting of single actions, that is, given BDI plan for the form $\phi[\alpha]\psi$, the body of the plan α consists of one primitive action. Such plans are akin to those in the GOAL agent programming language [de Boer *et al.*, 2007], as well as universal-plans [Schoppers, 1987], and reactive control modules [Baral and Son, 1998]. Let $\mathbf{\Pi}^{\mathcal{P}}_{Act}$ be the (infinite) set of all possible plan-rules given a set of actions *Act* and a set of domain propositions $\mathcal{P}$.

3.1 BDI-ATLES Syntax

The language of BDI-ATLES is defined over a finite set of atomic propositions $\mathcal{P}$, a finite set of agents $\mathcal{A}$, and a finite set of capability terms $\mathcal{C}$ available in the BDI application of concern. Intuitively, a capability term $c \in \mathcal{C}$ (e.g., Navigate) represents a plan library Π^c (e.g., Π^{Navigate}). As usual, a *coalition* is a set $A \subseteq \mathcal{A}$ of agents. A *capability assignment* ω consists of a set of pairs of agents along with their corresponding capabilities of the form $\langle agt : C_{agt}\rangle$, where $agt \in \mathcal{A}$ and $C_{agt} \subseteq \mathcal{C}$. A *goal assignment* ϱ defines the goal base (i.e., set of propositional formulas) for some agents. It consists of set of tuples of the form $\langle agt : G_{agt}\rangle$, where

agt $\in \mathcal{A}$ and G_{agt} is a set of boolean formulas over $\mathcal{P}$. $\mathcal{A}_\omega$ denotes the set of agents whose capabilities are defined by assignment ω, i.e., $\mathcal{A}_\omega = \{agt \mid \langle agt : C_{agt}\rangle \in \omega\}$. Set $\mathcal{A}_\varrho$ is defined analogously.

The set of BDI-ATLES formulas is then exactly like that of ATL(ES), except that coalition formulas are now of the form $\langle\!\langle A\rangle\!\rangle_{\omega,\varrho}\varphi$, where φ is a path formula (i.e., it is preceded by $\bigcirc$, $\Box$, or $\mathcal{U}$), A is a coalition, and ω and ϱ range over capability and goal assignments, respectively, such that $\mathcal{A}_\omega = \mathcal{A}_\varrho$. The intended meaning is as follows:

> $\langle\!\langle A\rangle\!\rangle_{\omega,\varrho}\varphi$ expresses that coalition of agents A can jointly force temporal condition φ to hold when BDI agents in $\mathcal{A}_\omega$ (or $\mathcal{A}_\varrho$) are equipped with capabilities as per ω and (initial) goals are as per ϱ.

Note that we require, in each coalition (sub)formula, that the agents for which capabilities and goals are assigned to be the same. This enforces the constraint that BDI-style agents have *both* plans and goals. So, a formula of the form $\langle\!\langle A\rangle\!\rangle_{\emptyset,\{\langle a_1:\{\gamma\}\rangle\}}\varphi$ is invalid, as agent a_1 has one goal (namely, to bring about γ), but its set of plans is undefined—we cannot specify what its rational behavior will be. ndition for player Ag is $\psi_{WIN} = G_B \wedge \mathsf{Ag}_B$: the player wins when collocated with gold at the depot.

3.2 BDI-ATLES Semantics

A BDI-ATLES *concurrent game structure* is a tuple $\mathcal{M} = \langle \mathcal{A}, Q, \mathcal{P}, Act, d, \mathcal{V}, \sigma, \Theta\rangle$, where:

- $\mathcal{A}$, Q, $\mathcal{P}$, *Act*, d, $\mathcal{V}$ and σ are as in ATL(ES).
- There is a distinguished dummy action NOOP $\in$ *Act* such that NOOP $\in d_{agt}(q)$ and $\sigma(q, \langle \text{NOOP}, \ldots, \text{NOOP}\rangle) = q$, for all *agt* $\in \mathcal{A}$ and $q \in Q$, i.e., NOOP is always available to all agents and the system remains still when all agents perform it.
- Capability function $\Theta : \mathcal{C} \mapsto \mathcal{F}(\mathbf{\Pi}^{\mathcal{P}}_{Act})$ maps capability terms to their (finite) set of plans. (Here, $\mathcal{F}(X)$ denotes the set of all finite subsets X.)

BDI-ATLES models are similar to ATLES ones, except that we use capability terms instead of strategy terms. In a nutshell, the challenge thus is to characterize what are the underlying "low-level" ATL strategies for agents with certain capabilities and goals. We call such strategies *rational strategies*, in that they are compatible with the standard BDI rational execution model [Rao and Georgeff, 1992]: *they represent the agent acting as per her available plans in order to achieve her goals in the context of her beliefs*.

So, given an agent *agt* $\in \mathcal{A}$, a plan-library Π, and a goal base $\mathcal{G}$, we define $\Sigma^{agt}_{\Pi,\mathcal{G}}$ as the set of standard ATL strategies for agent *agt* in $\mathcal{M}$ that are *rational strategies* when the agent is equipped with plan-library Π and has $\mathcal{G}$ as (initial) goals. These ATL strategies are such that the agent always chooses an action that is compatible with its available plans in order to achieve one of its goals in the context of its current beliefs. The intuition behind defining set $\Sigma^{agt}_{\Pi,\mathcal{G}}$ is to identify those "rational traces" in the structure that are compatible with the BDI deliberation process in accordance with the agent's goals and beliefs. Rational strategies, then, are those that only yield rational traces, which are in turn defined in three steps:

1. First, we define a *goal-marking* function $g(\lambda^+, i)$ denoting the "active" goal base of the agent at the i-th stage of trace λ^+.
2. Second, we define the set of indexes (i.e., stages) $Exec(\phi[\alpha]\psi, g, \lambda^+)$ in trace λ^+ where the plan $\phi[\alpha]\psi$ may have been executed by the agent: the plan's precondition ϕ was true, ψ was an active goal of the agent (as directed by goal-marking function g), and α was indeed performed.
3. Third, we define a rational trace λ^+ as one in which the agent has always executed one of its plans: for every index i, it is the case that $i \in Exec_{agt}(\phi[\alpha]\psi, g, \lambda^+)$, for some plan $\phi[\alpha]\psi$ in her know-how library.

Finally, we use $\Sigma^{agt}_{\Pi,\mathcal{G}}$ to denote the set of all ATL strategies whose executions always yield rational traces. The technical details of all the above steps and notions can be found in [Yadav and Sardina, 2012].

With the set of rational strategies $\Sigma^{agt}_{\Pi,\mathcal{G}}$ suitably defined, we are ready to detail the semantics for formulas of the form $\langle\!\langle A\rangle\!\rangle_{\omega,\varrho}\varphi$. Following ATLES, we first extend the notion of a joint strategy for a coalition to that of joint strategy *under a given capability and goal assignment*. So, given a capability (goal) assignment ω (ϱ) and an agent *agt* $\in \mathcal{A}_\omega$ (*agt* $\in \mathcal{A}_\varrho$), we denote *agt*'s capabilities (goals) under ω (ϱ) by $\omega[agt]$ ($\varrho[agt]$). Intuitively, an $\langle\omega, \varrho\rangle$*-strategy for coalition* A is a joint strategy for A such that *(i)* agents in $A \cap \mathcal{A}_\omega$ only follow "rational" (plan-goal compatible) strategies as per their ω-capabilities and ϱ-goals; and *(b)* agents in $A \backslash \mathcal{A}_\omega$ follow arbitrary strategies. Formally, an *$\langle\omega, \varrho\rangle$-strategy for coalition* A (with $\mathcal{A}_\omega = \mathcal{A}_\varrho$) is a collective strategy F_A for agents A such that for all $f_{agt} \in F_A$ with *agt* $\in A \cap \mathcal{A}_\omega$, it is the case that $f_{agt} \in \Sigma^{agt}_{\Pi,\mathcal{G}}$, where $\Pi = \cup_{c\in\omega[agt]}\Theta(c)$ and $\mathcal{G} = \varrho[agt]$. Note no requirements are asked on the strategies for the remaining agents $A \backslash \mathcal{A}_\omega$, besides of course being legal (ATL) strategies.

Using the notions of $\langle\omega, \varrho\rangle$-strategies and that of possible outcomes for a given collective strategy from ATL (refer to function *out*$(\cdot,\cdot)$ from Preliminaries), we are now able to state the meaning of BDI-ATLES (coalition) formulas:[2]

$\mathcal{M}, q \models \langle\!\langle A\rangle\!\rangle_{\omega,\varrho}\varphi$ *iff* there is a $\langle\omega, \varrho\rangle$-strategy F_A such that for all $\langle\omega, \varrho\rangle$-strategies $F_{\mathcal{A}_\omega \backslash A}$ for $\mathcal{A}_\omega \setminus A$, it is the case that $\mathcal{M}, \lambda \models \varphi$, for all paths $\lambda \in$ *out*$(q, F_A \cup F_{\mathcal{A}_\omega \backslash A})$.

Intuitively, F_A stands for the collective strategy of agents A guaranteeing the satisfaction of formula φ. Since F_A is a $\langle\omega, \varrho\rangle$-strategy, some agents in A—those whose capabilities and goals are defined by ω and ϱ, resp.—are to follow rational strategies. At the same time, because other agents outside the coalition could have also been assigned capabilities and goals, the chosen collective strategy F_A needs to work no matter how such agents (namely, agents $\mathcal{A}_\omega \setminus A$) behave, as long as they do it rationally given their capability and goal assignments. That is, F_A has to work with *any* rational collective strategy $F_{\mathcal{A}_\omega \backslash A}$. Finally, the behavior of all remaining

[2] As with ATL(ES), φ ought to be a path formula and is interpreted in the usual manner. We omit the other ATL-like cases for brevity; see [Walther *et al.*, 2007].

foreach φ' *in Sub(φ) w.r.t.* $\mathcal{M} = \langle \mathcal{A}, Q, \mathcal{P}, Act, d, \mathcal{V}, \sigma, \Theta \rangle$ **do**
  case $\varphi' = p : [\varphi']_{\mathcal{M}} = \mathcal{V}(p)$;
  case $\varphi' = \neg\theta : [\varphi']_{\mathcal{M}} = ([\text{TRUE}]_{\mathcal{M}} \setminus [\theta]_{\mathcal{M}})$;
  case $\varphi' = \theta_1 \vee \theta_2 : [\varphi']_{\mathcal{M}} = [\theta_1]_{\mathcal{M}} \cup [\theta_2]_{\mathcal{M}}$;
  case $\varphi' = \langle\langle A \rangle\rangle_{\omega,\varrho} \bigcirc \theta :$
  $[\varphi']_{\mathcal{M}} = ws(Pre(A, \omega, \Theta, [\theta]_{\mathcal{M}_\varrho}) \cap [\![\varrho]\!])$;
  case $\varphi' = \langle\langle A \rangle\rangle_{\omega,\varrho} \Box\theta : \rho = [\text{TRUE}]_{\mathcal{M}_\varrho}; \tau = [\theta]_{\mathcal{M}_\varrho}$;
    while $\rho \not\subseteq \tau$ **do** $\rho = \tau$; $\tau = Pre(A, \omega, \Theta, \rho) \cap [\theta]_{\mathcal{M}_\varrho}$ **od;**
    $[\varphi']_{\mathcal{M}} = ws(\rho \cap [\![\varrho]\!])$;
  case $\varphi' = \langle\langle A \rangle\rangle_{\omega,\varrho} \theta_1 \mathcal{U} \theta_2 : \rho = [\text{FALSE}]_{\mathcal{M}_\varrho}; \tau = [\theta_2]_{\mathcal{M}_\varrho}$;
    while $\tau \not\subseteq \rho$ **do** $\rho = \rho \cup \tau; \tau = Pre(A, \omega, \Theta, \rho) \cap [\theta_1]_{\mathcal{M}_\varrho}$ **od;**
    $[\varphi']_{\mathcal{M}} = ws(\rho \cap [\![\varrho]\!])$;
od;
return $[\varphi']_{\mathcal{M}}$;

Figure 1: BDI-ATLES symbolic model checking.

agents—namely those in $\mathcal{A} \setminus (A \cup \mathcal{A}_\omega)$—are taken into account when considering all possible outcomes, after all strategies for agents in $A \cup \mathcal{A}_\omega$ have been settled.

4 BDI-ATLES Model Checking

In order to model checking a formula $\langle\langle A \rangle\rangle_{\omega,\varrho}\varphi$ against a BDI-ATLES model $\mathcal{M}$, one has to consider the rational choices of each BDI agent. These rational choices are the consequence of the agent's goals and capabilities specified by functions ϱ and ω in formula. The core idea, then, is to restrict the options of the BDI agents at each step relative to their applicable plans. Next, we extend the model $\mathcal{M}$ to embed the possible goals (based on the goal assignment) of BDI agents into each state, then we discuss the model checking algorithm and its complexity.

Given a BDI-ATLES model $\mathcal{M}=\langle \mathcal{A}, Q, \mathcal{P}, Act, d, \mathcal{V}, \sigma, \Theta \rangle$ and a goal assignment ϱ, the *goal-extended model* is a tuple $\mathcal{M}_\varrho=\langle \mathcal{A}, Q_\varrho, \mathcal{P}, Act, d_\varrho, \mathcal{V}_\varrho, \sigma_\varrho, \Theta \rangle$, where:

- $Q_\varrho \subseteq Q \times \prod_{agt \in \mathcal{A}_\varrho} 2^{\varrho[agt]}$ is the set of extended states, now accounting for the possible goals of BDI agents. When $q_\varrho = \langle q, g_1, \ldots, g_{|\mathcal{A}_\varrho|} \rangle \in Q_\varrho$, where $q \in Q$ and $g_i \subseteq \varrho[agt_i]$, is an extended state, we use $ws(q_\varrho) = q$ and $gl(agt_i, q_\varrho) = g_i$ to project $\mathcal{M}$'s world state and agt_i's goals. To enforce belief-goal consistency we require no agent ever wants something already true: there are no $q_\varrho \in Q_\varrho$, $agt \in \mathcal{A}_\varrho$, and formula γ such that $\mathcal{V}(ws(q_\varrho)) \models \gamma$ and $\gamma \in gl(agt, q_\varrho)$.
- $\mathcal{V}_\varrho(q_\varrho) = \mathcal{V}(ws(q_\varrho))$ and $d_\varrho(agt, q_\varrho) = d(agt, ws(q_\varrho))$, for all $q_\varrho \in Q_\varrho$, that is, state evaluation and physical executability remains unchanged.
- $\sigma_\varrho(q_\varrho, \vec{a}) = \langle q', g'_1, \ldots, g'_{|\mathcal{A}_\varrho|} \rangle$, where $q' = \sigma(ws(q_\varrho), \vec{a})$ and $g'_i = gl(agt_i, q_\varrho) \setminus \{\gamma \mid \gamma \in gl(agt_i, q_\varrho), \mathcal{V}(q') \models \gamma\}$, is the transition function.

States of model $\mathcal{M}_\varrho$ are similar to that of $\mathcal{M}$, except that they also include possible agent's goals. Observe that the transition relation caters for persistence of goals as well as dropping of achieved goals. Hence, the transition relation is well-defined within Q_ϱ states. Interestingly, though, the extended model ensures rationality with respect to goals is maintained, it does not restrict the original physical executability of actions. Therefore, the extended model accommodates both rational and irrational paths. However, it is now possible to distinguish between them, as one can reason about applicable plans in each extended state.

As standard, $[\varphi]_{\mathcal{M}}$ denotes the set of states in $\mathcal{M}$ satisfying the formula φ. We extend $ws(\cdot)$ projection function to sets of extended states, that is, $ws(S) = \bigcup_{q \in S}\{ws(q)\}$. Thus, $ws([\varphi]_{\mathcal{M}_\varrho})$ denotes the set of all world states in $\mathcal{M}$ that are part of an extended state in $\mathcal{M}_\varrho$ satisfying the formula φ. Also, $[\![\varrho]\!]$ denotes the set of extended states where the agents' goals are as per goal assignment ϱ; formally, $[\![\varrho]\!] = \{q \mid q \in Q_\varrho, \forall agt \in \mathcal{A}_\varrho : gl(agt, q) = \varrho[agt]\}$.

Figure 1 shows the model checking algorithm for BDI-ATLES. It is based on the symbolic model checking algorithm for ATL [Alur *et al.*, 2002] and ATLES [Walther *et al.*, 2007]. The first three cases are handled in the same way as in ATL(ES). To check the BDI-ATLES coalition formulae $\langle\langle A \rangle\rangle_{\omega,\varrho}\varphi$, we extend the model as above (relative to the formula's goal assignment ϱ), and then check the plain ATL coalition formula $\langle\langle A \rangle\rangle\varphi$ in such extended model. Note that only the set of states having the goals as per the initial goal assignment are returned— initial goals of all agents are active in the first state of any rational trace. Unlike standard ATL model checking, we restrict the agents' action choices as per their capabilities. We do this by modifying the usual pre-image function $Pre(\cdot)$ to only take into account actions compatible with agents' applicable plans. Intuitively, $Pre(A, \omega, \Theta, \rho)$ is the set of (extended) states from where agents in coalition A can jointly force the next (extended) state to be in set ρ no matter how all other agents (i.e., agents in $\mathcal{A} \setminus A$) may act and provided all BDI-style agents (i.e., agents with capabilities defined under ω and Θ) behave rationally. Due to lack of space we omit the formal definition of *Pre*, which can be found in [Yadav and Sardina, 2012].

It is easy to see that the extended model is, in general, exponentially larger than the original one with respect to the number of goals $\max_{agt \in \mathcal{A}}(|\varrho[agt]|)$ and agents $|\mathcal{A}_\varrho|$. A more promising case arises when goals are given a *reactive maintenance* interpretation [Duff *et al.*, 2006; Dastani *et al.*, 2006]—conditions that ought to be restored whenever "violated". Interestingly, under such interpretation of goals, we retain ATL(ES) polynomial complexity, which is of course a tight bound as the resulting framework subsumes ATL (just take $\omega = \varphi = \emptyset$) and model checking ATL is PTIME-complete [Alur *et al.*, 2002]. See [Yadav and Sardina, 2012] for details.

5 Conclusion

We have proposed a framework for verifying BDI-type agents using an ATL-like logic, thus bringing together work in verification of strategic behaviour and agent programming.

Among other things, we would like to extend our work to more general agent programs as well as to investigate how to integrate plausibility reasoning [Jamroga and Bulling, 2007] in our logic to focus the verification to certain parts of an ATL structure using more declarative specifications.

References

[Alechina *et al.*, 2007] N. Alechina, M. Dastani, B. S. Logan, and John-Jules Meyer. A logic of agent programs. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 795–800, 2007.

[Alechina *et al.*, 2008] N. Alechina, M. Dastani, B. S. Logan, and John-Jules Meyer. Reasoning about agent deliberation. In *Proc. of Principles of Knowledge Representation and Reasoning (KR)*, pages 16–26, 2008.

[Alur *et al.*, 2002] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, (49):672–713, 2002.

[Baral and Son, 1998] Chitta Baral and Tran Cao Son. Relating theories of actions and reactive control. *Electronic Transactions of AI (ETAI)*, 2(3–4):211–271, 1998.

[Bordini *et al.*, 2006] R. H. Bordini, L. Braubach, M. Dastani, A. Fallah-Seghrouchni, J. J. Gómez Sanz, J. Leite, G. O'Hare, A. Pokahr, and A. Ricci. A survey of programming languages and platforms for multi-agent systems. *Informatica (Slovenia)*, 30(1):33–44, 2006.

[Bratman *et al.*, 1988] M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(3):349–355, 1988.

[Busetta *et al.*, 1999] P. Busetta, R. Rönnquist, A. Hodgson, and A. Lucas. JACK intelligent agents: Components for intelligent agents in Java. *AgentLink Newsletter*, 2:2–5, January 1999.

[Dastani and Jamroga, 2010] M. Dastani and W. Jamroga. Reasoning about strategies of multi-agent programs. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 997–1004, 2010.

[Dastani *et al.*, 2006] Mehdi Dastani, Birna van Riemsdijk, and John-Jules Meyer. Goal types in agent programming. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1285–1287, 2006.

[de Boer *et al.*, 2007] F.S. de Boer, K.V. Hindriks, W. van der Hoek, and J.J.C. Meyer. A verification framework for agent programming with declarative goals. *Journal of Applied Logic*, 5(2):277–302, 2007.

[Duff *et al.*, 2006] Simon Duff, James Harland, and John Thangarajah. On proactivity and maintenance goals. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1033–1040, 2006.

[Jamroga and Bulling, 2007] W. Jamroga and N. Bulling. A framework for reasoning about rational agents. In *Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1–3, 2007.

[Padgham and Lambrix, 2005] L. Padgham and P. Lambrix. Formalisations of capabilities for BDI-agents. *Autonomous Agents and Multi-Agent Systems*, 10(3):249–271, May 2005.

[Rao and Georgeff, 1992] Anand S. Rao and Michael P. Georgeff. An abstract architecture for rational agents. In *Proc. of Principles of Knowledge Representation and Reasoning (KR)*, pages 438–449, 1992.

[Schoppers, 1987] M. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1039–1046, 1987.

[Walther *et al.*, 2007] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In *Conference on Theoretical Aspects of Rationality and Knowledge*, pages 269–278. ACM Press, 2007.

[Yadav and Sardina, 2012] Nitin Yadav and Sebastian Sardina. Reasoning about BDI agent programs using ATL-like logics. In *Proc. of the European Conference on Logics in Artificial Intelligence (JELIA)*, pages 437–449, 2012. Extended version available from CoRR arXiv at http://arxiv.org/abs/1207.3874.