CTL AgentSpeak(L): a Specification Language for Agent Programs

Alejandro Guerra-Hernández[a], José Martín Castro-Manzano[a], Amal El-Fallah-Seghrouchni[b]

[a] *Departamento de Inteligencia Artificial*
Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Sebastián Camacho No. 5, Xalapa, Ver., México, 91000
[b] *Laboratoire d'Informatique de Paris 6*
Université Pierre et Marie Curie
Avenue du Président Kennedy, Paris, France, 75016

Abstract

This work introduces *CTL AgentSpeak(L)*, a logic to specify and verify expected properties of rational agents implemented in the well known agent oriented programming language *AgentSpeak(L)*. Our approach is closely related to the BDI_{CTL} multi-modal logic, used to reason about agents in terms of their beliefs (B), desires (D), intentions (I), and the temporal logic CTL. A new interpretation for the temporal operators, grounded in the transition system induced by the operational semantics of *AgentSpeak(L)*, is proposed. The main contribution of the approach is a better understanding of the relation between the programming language and its logical specification, enabling us to prove expected or desired properties for any agent programmed in the language, e.g., commitment strategies. The results, as well as the specification language proposed, are very useful to reconcile computational and philosophical aspects of practical reasoning, e.g., approaching single-minded commitment as a policy-based reconsideration case.

Key words: AgentSpeak(L), Specification, Commitment, Reconsideration

1. Introduction

The theory of practical reasoning proposed by Bratman [3, 4], expounds the philosophical foundation for the computational approaches to rational agency, known as BDI (Belief-Desire-Intention) systems. This theory is innovative because it does not reduce intentions to some combination of beliefs and desires, but indeed it assumes that intentions are composed by hierarchical, partial plans. Such assumption explains better temporal aspects of practical reasoning as future intentions, persistence, reconsideration and commitment. Different multi-modal BDI logics [15, 17, 18] have been proposed to formally characterize the rational behavior of such agents, in terms of the properties of the intentional attitudes and the relations among them. For instance, blind, single-minded, and open-minded commitment strategies [13] define when it is rational for a class of agents to drop an intention. Due to their expressiveness, these logics are used to reason about the agents properties, e.g., if an agent is blind or single-minded committed; but they are not used to program them, because of their computational cost.

Agent oriented programming languages, such as *AgentSpeak(L)* [14], have been proposed to reduce the gap between the theory (logical specification) and practice (implementation) of rational agents. Bordini et al. [2] have completed the operational semantics for this language, and implemented an interpreter for it, known as Jason. Even when this programming language has a well defined operational semantics, the verification of rational properties of the implemented agents is not evident, since intentional and time modalities are abandoned for the sake of efficiency. Questions like, what kind of commitment strategy is used by the *AgentSpeak(L)* agents? require a specification and verification formalism in order to be answered. Furthermore, a formalism grounded in the operational semantics of the programming language, is required to reason about such rational properties of the implemented agents.

Email addresses: aguerra@uv.mx (Alejandro Guerra-Hernández), Amal.Elfallah@lip6.fr (Amal El-Fallah-Seghrouchni)

We propose *CTL AgentSpeak(L)* as a logic for specification and verification of *AgentSpeak(L)* agents. The approach proposed is similar to the BDI_{CTL} [15] logic, defined as a $B^{KD45}D^{KD}I^{KD}$ modal system, with temporal operators defined after the computational tree logic (CTL) [7], but redefines the semantics of the intentional and temporal operators in order to ground them. The redefinition of the semantics for the CTL temporal operators in terms of a Kripke structure, induced by the transition system defining the operational semantics of *AgentSpeak(L)* is the main technical contribution of the paper. The semantics of the intentional operators (*BDI*) is adopted from the work of Bordini et al. [1]. As a result, the semantics of *CTL AgentSpeak(L)* is grounded in the operational semantics of programming language. In this way, we can prove if any agent programmed in *AgentSpeak(L)* satisfies certain properties expressed in the logical specification.

This work was motivated by a question related to rational agents, learning, and commitment: can a single-minded committed behavior [13] be approached as a policy-based reconsideration [3] where the policies are intentionally learnt by the agents [8, 9, 11]? The question is relevant, because it reconciles the computational concept of commitment (the blind, singled-minded, and open-minded strategies) with the philosophical concept of reconsideration (all reconsideration of α opens the question of whether α), through an adaptive computation of the policy of reconsideration (intentional learning). Because of this, the specification approach proposed here is exemplified verifying the commitment strategies for *AgentSpeak(L)* to answer what kind of strategy is adopted by such agents.

The paper is organized as follows. Section 2 introduces the commitment strategies as defined using BDI_{CTL} [15] and discusses briefly the reconsideration cases identified by Bratman [3]. Section 3 defines the syntax and semantics of the agent oriented programming language *AgentSpeak(L)*. The operational semantics defined in this section will be used to define the semantics of the formalism proposed in this work. Section 4 presents the main contribution of the paper, the definition of *CTL AgentSpeak(L)*, a specification and verification formalism for *AgentSpeak(L)*. Section 5 shows how the commitment strategies introduced in section 2 can be verified under *CTL AgentSpeak(L)*. Section 6 offers concluding remarks and discusses future work.

2. Commitment

As mentioned previously, different computational theories have been proposed to capture the theory of Bratman [3] on intentions, plans and practical reasoning. The foundational work of Cohen and Levesque [5], for example, defined intention as a combination of belief and desire based on the concept of persistent goal. A critical analysis of this theory [16] showed that it failed to capture important aspects of commitment. Alternatively, commitment has been approached as a process of maintenance and revision of intentions, relating current and future intentions. Under the latter approach, different types of commitment strategies define different types of agents. Three of them have been extensively studied in the context of BDI_{CTL} [15], where CTL denotes computational tree logic [7], the well known temporal logic:

- **Blind commitment.** An agent intending that inevitably (A) eventually ($\Diamond$) is the case that ϕ, inevitably maintains his intentions until (U) he actually believes ϕ:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi)) \tag{1}$$

- **Single-mind commitment.** An agent maintains his intentions as long as he believes they are not achieved or optionally (E) they are eventually achievable:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)) \tag{2}$$

- **Open-mind commitment.** An agent maintains his intentions as long as they are not achieved or they are still desired:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{DES}(\mathsf{A}\Diamond\phi))) \tag{3}$$

ag	::=	$bs \quad ps$		at	::=	$P(t_1, \ldots, t_n)$	$(n \geq 0)$
bs	::=	$b_1 \ldots b_n$	$(n \geq 0)$	a	::=	$A(t_1, \ldots, t_n)$	$(n \geq 0)$
ps	::=	$p_1 \ldots p_n$	$(n \geq 1)$	g	::=	$!at \mid ?at$	
p	::=	$te : ct \leftarrow h$		u	::=	$+b \mid -b$	
te	::=	$+at \mid -at \mid +g \mid -g$					
ct	::=	$ct_1 \mid \top$					
ct_1	::=	$at \mid \neg at \mid ct_1 \wedge ct_1$					
h	::=	$h_1; \top \mid \top$					
h_1	::=	$a \mid g \mid u \mid h_1; h_1$					

Table 1: Sintax of *AgentSpeak(L)*. Adapted from Bordini et al. [2]

For example, a blind agent (eq. 1) intending eventually to go to Paris, will maintain his intention, for any possible course of action (inevitable), until he believes he is going to Paris. A single-minded agent (eq. 2) can drop such intention if he believes it is not possible anymore to go to Paris, e.g., given a certain policy of reconsideration defining, for instance, that such intention should be dropped if the plane ticket is too expensive; or more properly, believing that if the plane ticket is to expensive then it is not eventually possible going to Paris. An open-minded agent (eq. 3) can drop the intention if he does not desire anymore going to Paris.

Then, an interesting question is what kind of commitment strategy is followed by *AgentSpeak(L)* agents? Furthermore, how can we relate commitment with policy-based reconsideration in *AgentSpeak(L)*? Bratman (see [3], pp. 60–75) argues that there are three cases of reconsideration in practical reasoning. Non-reflective reconsideration has short effects, while deliberative one is very expensive. Policy-based reconsideration is a compromise between impact and cost. Obviously, if an agent is blindly committed, we can not talk about any form of reconsideration. But if this is not the case, single-minded commitment could be approached as a policy-based reconsideration, where policies are computed through intentional learning [8, 9, 10, 11]. In this way we would reconcile a relevant aspect of the computational theories of BDI agency (commitment) with its philosophical foundation (reconsideration *à la* Bratman). Answering these questions was the main motivation of this work.

3. *AgentSpeak(L)*

The syntax and semantics of *AgentSpeak(L)* [14] adopted in this work, are defined after the specification for its interpreter Jason [2]. The operational semantics of the language is given in terms of a transition system Γ that, being a Kripke structure, is used to define the semantics of the intentional and temporal operators in the proposed specification language *CTL AgentSpeak(L)*.

3.1. Syntax

The syntax of *AgentSpeak(L)* is shown in table 1. As usual, an agent ag is formed by a set of beliefs bs (grounded literals) and plans ps. Each plan $p \in ps$ has the form $triggerEvent : context \leftarrow body$. The context of a plan is a literal or a conjunction of them. A non empty plan body is a finite sequence of actions, goals (achieve or test an atomic formula), and beliefs updates (addition or deletion). $\top$ denotes empty elements, e.g., contexts, plan bodies, and intentions. The addition or deletion of beliefs and goals are known as trigger events.

For instance, table 2 shows an *AgentSpeak(L)* program for an agent called `bob` who has as an achieve goal (!) to print the factorial of five (line 7). The agent `bob` initially believes that the factorial of zero is one (line 4). There are three plans in the library of agent `bob` to accomplish his goal: `p1`, `p2`, and `p3` (plan labels defined with `@` are optional). The plan `p1` (line 11) is relevant when an achieve goal to print the factorial of a number is added, as defined in its trigger event. Since the context of this plan is empty, it is always applicable. The body of `p1` has two steps: the achieve goal to compute the factorial (line 11) and an action to print the result (line 12). The plans `p2` and `p3` (lines 16 and 20, respectively) compute the factorial of a given number. The first one is only relevant when $N = 0$, as defined in its trigger event. Its body include a test goal (?) to compute the factorial of zero (line 17). The second plan, as expressed by its context, is applicable for $N > 0$. The output in the console is "[bob] Factorial of 5 is 120".

In what follows, this example will be used to illustrate the concepts of the operational semantics for *AgentSpeak(L)* that are relevant for the purposes of this paper.

```
// Agent bob in project factorial.mas2j

/* Initial beliefs and rules */
factorial(0,1).

/* Initial goals */
!print_factorial(5).

/* Plans */
@p1
+!print_factorial(N) <-
   !factorial(N,F);
   .print("Factorial of ",N," is ",F).

@p2
+!factorial(0,F) <-
   ?factorial(0,F).

@p3
+!factorial(N,F) : N > 0 <-
   !factorial(N-1,F1);
   F = F1 * N.
```

Table 2: An *AgentSpeak(L)* program that computes the factorial of five (slightly modified from Bordini et al. [2], p. 12).

3.2. Operational semantics

The operational semantics of *AgentSpeak(L)* is defined in terms of a transition system Γ between configurations. A configuration $\langle ag, C, M, T, s\rangle$ is composed of:

- ag is a tuple $\langle bs, ps\rangle$ formed by the beliefs bs and plans ps of the agent, as defined in the agent program.
- An agent circumstance C is a tuple $\langle I, E, A\rangle$ where I is a set of intentions $\{i_1, i_2, \ldots, i_n\}$ s.t. each $i \in I$ is a stack of partially instantiated plans $p \in ps$. The operator $\ddagger$ is used to separate elements in a stack. $[\alpha \ddagger \beta]$ is a stack with two elements, α at its top; E is a set of events $\{\langle te_1, i_1\rangle, \langle te_2, i_2\rangle, \ldots, \langle te_n, i_n\rangle\}$, s.t. each te is a $triggerEvent$ and each i is a non empty intention (internal event) or an empty intention $\top$ (external event); and A is a set of actions to be executed by the agent in the environment.
- M is a tuple $\langle In, Out, SI\rangle$ where In is the mailbox of the agent, Out is a list of messages to be delivered and SI is a register of suspended intentions (intentions that wait for an answer message to be resumed). It is not relevant for the purposes of this paper.
- T is a tuple $\langle R, Ap, \iota, \epsilon, \rho\rangle$ where R is the set of relevant plans given certain event; Ap is the set of applicable plans (the subset of plans $p \in R$ s.t. $bs \models Ctxt(p)$, where the function $Ctxt$ return the context of a given plan or *true* if such context is empty); ι, ϵ y ρ are, respectively, the intention, event, and plan currently being considered while reasoning.
- The label $s \in \{SelEv, RelPl, AppPl, SelAppl, SelInt, AddIM, ExecInt, ClrInt, ProcMsg\}$ indicates the current step in the reasoning cycle of the agent.

Transitions are defined in terms of semantic rules of the form:

$$(\textbf{rule id}) \quad \frac{cond}{C \rightarrow C'}$$

where $C = \langle ag, C, M, T, s\rangle$ is a configuration that can be transformed into a new one C', if $cond$ holds.

Appendix A includes the semantic rules that are relevant for the purposes of this paper. Since communication and speech acts are out of the scope of this work, message processing rules have been omitted for simplicity. Figure 1 shows the transition system Γ induced by these semantic rules. States are labeled with possible values for s. Transitions correspond to the semantic rules identifiers. The reasoning cycle of an agent starts at $s = ProcMsg$, processing messages and updating the perception, which adds events to C_E (C_α denotes the element α of circumstance C, the same for other elements of a configuration). Then, at $s = SelEv$, one of these events is selected ($\mathcal{S}_\mathcal{E}(C_E) = \langle te, i\rangle$). The selection functions for events ($\mathcal{S}_\mathcal{E}$), intentions ($\mathcal{S}_\mathcal{I}$), and applicable plans ($\mathcal{S}_\mathcal{O}$) are assumed fair. By default, these functions have a first-input-first-output behavior; but they can be redefined by the programmer, so that fairness must be considered if this is the case.

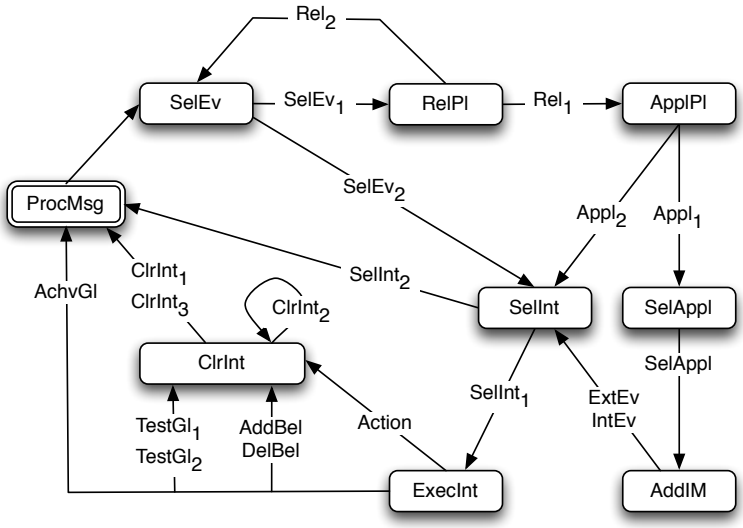

Figure 1: The interpreter for *AgentSpeak(L)* as a transition system.

Once an event $\langle te, i\rangle$ is selected, following $SelEv_1$ leads to the computation of a set of relevant plans at $s = RelPl$, accordingly to definition 1. If this set is empty, by the transition Rel_2 a new event is selected. If there is not any event in C_E, by transition $SelEv_2$ an intention is selected in order to be executed.

Definition 1 (Relevant plans). *Given a set of plans ag_{ps}, the subset of relevant plans for a selected event $\langle te, i\rangle \in C_E$, is defined as:*

$$RelPlans(ps, te) = \{(p, \theta) | p \in ps \wedge \theta = mgu(te, TrEv(p))\}$$

where $TrEv(te' : context \leftarrow body) = te'$ gets the trigger event of a plan, C_E denotes the events E in a given circumstance C, and mgu computes the most general unifier.

For instance, the factorial agent program (see table 2) will post an event when perceiving the achieve goal of printing the factorial of five, s.t. $C_E = \{\langle +!print\_factorial(5), \top\rangle\}$. The computed set of relevant plans for such an event is $T_R = \{(p1, \theta)\}$, where $\theta = \{N/5\}$. Then, once a non empty set of relevant plans T_R has been computed at $s = RelPl$, following the transition Rel_1 leads to the computation of a set of applicable plans T_{Ap} at $s = AppPl$, as stated in the following definition:

Definition 2 (Applicable plans). *Given a set of relevant plans T_R and a set of beliefs ag_{bs}, the subset of applicable plans is defined as:*

$$AppPlans(bs, R) = \{(p, \theta\theta') | (p, \theta) \in T_R \wedge \theta' \text{ is s.t. } bs \models Ctxt(p)\theta\theta'\}$$

where θ' is the substitution computed when verifying if the context of relevant plan p, affected by its relevant substitution θ, is a logical consequence of the beliefs of the agent bs. $Ctxt(p)$ returns the context of plan p or true, if such context is empty.

Following the factorial example, since the context of plan p1 is empty, then $Ctxt(p1) = true$ which is verified to be a logical consequence of the belief of the agent ag_{bs} with an associated empty substitution $\theta' = \epsilon$, so that the set of applicable plans in this case is $T_{Ap} = \{p1, \theta\theta'\}$, where the substitution composition $\theta\theta' = N/5$.

Then the agent proceeds selecting an applicable plan to form an intention (following $Appl_1$ to arrive at $SelAppl$) or, if no applicable plans were found, selecting an intention to be executed (following $Appl_2$ to arrive at $SelInt_1$). The execution of an intention changes the environment and the mental attitudes of the agent (dropping accomplished plans and intentions at *ClrInt* included). *ProcMsg* generates new events, and so on.

Although the operational semantics of *AgentSpeak(L)* clearly defines the practical reasoning performed by an agent, it is difficult to prove general rational properties for the implemented agents. For instance, it is difficult to verify which commitment strategy (see eqs. 1, 2, 3) is followed by these agents. This is due to the abandon of intentional and temporal modalities in *AgentSpeak(L)*, the main reason to propose *CTL AgentSpeak(L)* for the specification and verification of such properties.

4. *CTL AgentSpeak(L)*

CTL AgentSpeak(L) is closely related to BDI_{CTL} [13, 15]. The main idea is to redefine the semantics of temporal and intentional operators in terms of the operational semantics of the programming language. This grounds the semantics to reason about particular kinds of agents, in this case *AgentSpeak(L)* agents. Once this has been accomplished, we can use the logic to reason about general properties of such agents. Similar approaches have been explored for other instances of agent oriented programming languages, e.g, a simplified version of 3APL [6]. In what follows, the syntax and semantics of the proposed formalism are defined.

4.1. Syntax

Well-formed formulae (wff) in *CTL AgentSpeak(L)* include intentional and temporal expressions:

Definition 3 (intentional wff). *If ϕ is an atomic grounded formula in* AgentSpeak(L)*, then ϕ,* $\mathsf{BEL}(\phi)$*,* $\mathsf{DES}(\phi)$*, and* $\mathsf{INTEND}(\phi)$ *are* CTL AgentSpeak(L) *wff.*

Definition 4 (temporal wff). *Temporal wff are divided, as usual, in state and path formulae. State wff are defined as:*

s1 *Every intentional wff is a state formula.*

s2 *If ϕ and ψ are state formulae, $\phi \wedge \psi$ and $\neg\phi$ are state formulae.*

s3 *If ϕ is a path wff, then $\mathsf{E}\phi$ (optional) and $\mathsf{A}\phi$ (inevitable) are state formulae.*

Path wff are defined as:

p1 *Every state wff is also a path wff.*

p2 *If ϕ and ψ are path wff, then $\neg\phi$, $\phi \wedge \psi$, $\bigcirc\phi$ (next), $\Diamond\phi$ (eventually), and $\phi\ \mathsf{U}\ \psi$ (until) are path wff.*

For example, $\mathsf{INTEND}(\mathsf{A}\Diamond go(paris))\ \mathsf{U}\ \neg\mathsf{BEL}(go(paris, summer)$, expressing that the agent will intend inevitably (A) for every course of action, eventually ($\Diamond$) going to Paris until (U) he does not believe go to Paris in summer, is a well formed formula.

4.2. Semantics

The semantics of the intentional operators BEL, DES and INTEND is adopted from Bordini et al. [1]. First an auxiliary function for getting the atoms (at) subject of an achieve goal in a given intention ($+!\ at$), is defined:

$$
\begin{aligned}
agoals(\top) &= \{\}, \\
agoals(i[p]) &= \begin{cases} \{at\} \cup agoals(i) & \text{if } p = +!\ at : context \leftarrow body, \\ agoals(i) & \text{otherwise} \end{cases}
\end{aligned}
$$

Definition 5 (intentional semantics). *The semantics of the* BEL*,* DES*, and* INTEND *operators for a given agent ag and its circumstance C in an* AgentSpeak(L) *configuration is:*

$$\mathsf{BEL}_{\langle ag,C\rangle}(\phi) \equiv ag_{bs} \models \phi \tag{4}$$

$$\mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \equiv \phi \in \bigcup_{i\in C_I} agoals(i) \vee \bigcup_{\langle te,i\rangle\in C_E} agoals(i) \tag{5}$$

$$\mathsf{DES}_{\langle ag,C\rangle}(\phi) \equiv \langle +!\phi, i\rangle \in C_E \vee \mathsf{INTEND}(\phi) \tag{6}$$

If the agent ag and his circumstance C are explicit, we simply write $\mathsf{BEL}(\phi)$, $\mathsf{DES}(\phi)$, and $\mathsf{INTEND}(\phi)$. So an agent ag is said to believe the atomic formula ϕ, if ϕ is a logical consequence of the beliefs bs of ag. An agent is said to intend the atomic formula ϕ, if ϕ is the subject of an achieve goal in the active intentions of the agent (C_I) or in his suspended intentions associated to events to be processed (C_E). An agent is said to desire the atomic formula ϕ, if there is an event in C_E which trigger is an achieve goal about ϕ or if ϕ is intended.

The semantics of the temporal operators: next($\bigcirc$), eventually ($\Diamond$), and until (U), as well as the path quantifier inevitable (A), required a Kripke structure $\langle S, R, L\rangle$ where S is a set of states, R is a serial relation on $S \times S$ and L is a labelling or a valuation function for formulae in the states. The transition system Γ (see figure 1), defining the operational semantics of *AgentSpeak(L)*, induces a Kripke structure:

Definition 6 (*AgentSpeak(L)* Kripke structure). *$K = \langle S, R, V\rangle$ is a Kripke structure induced by the transition system Γ which defines the operational semantics of* AgentSpeak(L)*, where:*

- *S is a set of agent configurations $\langle ag, C, M, T, s\rangle$.*
- *$R \subseteq S^2$ is a serial relation s.t. for all pair of configurations $(c_i, c_j) \in R$, it follows that $\Gamma(c_i) = c_j$.*
- *V is a valuation function over the intentional and temporal operators, defined after their semantics (see definitions 5 and 7), e.g., $V_{\mathsf{BEL}}(c, \phi) \equiv \mathsf{BEL}(\phi)$ at the configuration $c = \langle ag, C, M, T, s\rangle$, etc.*

As usual, $x = c_0, \ldots, c_n$ denotes a path in the Kripke structure, i.e., a sequence of configurations s.t. for all $c_i \in S$, $(c_i, c_{i+1}) \in R$. The expression x^j denotes the suffix of path x starting at configuration c_i.

Definition 7 (temporal semantics). *The semantics of the state wff is defined for a given current configuration $c_i \in K_S$:*

$$\begin{aligned}
K, c_i \models \mathsf{BEL}(\phi) &\Leftrightarrow \phi \in V_{\mathsf{BEL}}(c_i, \phi)\\
K, c_i \models \mathsf{INTEND}(\phi) &\Leftrightarrow \phi \in V_{\mathsf{INTEND}}(c_i, \phi)\\
K, c_i \models \mathsf{DES}(\phi) &\Leftrightarrow \phi \in V_{\mathsf{DES}}(c_i, \phi)\\
K, c_i \models \mathsf{E}\phi &\Leftrightarrow \exists x^i \exists c_{j\geq i} \in x^i \text{ s.t. } K, c_j \models \phi\\
K, c_i \models \mathsf{A}\phi &\Leftrightarrow \forall x^i \exists c_{j\geq i} \in x^i \text{ s.t. } K, c_j \models \phi
\end{aligned}$$

The semantic of the path formulae is defined as follows:

$$\begin{aligned}
K, c_i \models \phi &\Leftrightarrow K, x^i \models \phi, \text{where } \phi \text{ is a state wff}\\
K, c_i \models \bigcirc\phi &\Leftrightarrow K, x^{i+1} \models \phi\\
K, c_i \models \Diamond\phi &\Leftrightarrow \exists c_{j\geq i} \in x^i \text{ s.t. } K, c_j \models \phi\\
K, c_i \models \phi \,\mathsf{U}\, \psi &\Leftrightarrow (\exists c_{k\geq i} \text{ s.t. } K, x^k \models \psi \wedge \forall c_{i\leq j<k} \quad K, x^j \models \phi) \vee\\
&\qquad (\exists c_{j\geq i} \quad K, x^j \models \phi).
\end{aligned}$$

Observe that the semantics of until corresponds to weak until (ψ can never occur). Once satisfaction over state and path formulae has been defined, we can define satisfaction and validity over *AgentSpeak(L)* runs.

Definition 8 (Run). *Given an initial* AgentSpeak(L) *configuration c_0, the run K^0_Γ denotes the sequence of configurations $c_0, c_1, \ldots$ such that $\forall i \geq 1, c_i = \Gamma(c_{i-1})$.*

Definition 9 (Satisfaction over runs). *Given an* AgentSpeak(L) *run K^0_Γ the property $\phi \in$* CTL AgentSpeak(L) *is satisfied, if $\forall i \geq 0, K^0_\Gamma, c_i \models \phi$.*

Definition 10 (Validity). *A property $\phi \in$* CTL AgentSpeak(L) *is valid, if for any initial configuration, $K^0_\Gamma, c_0 \models \phi$, e.g., if it is satisfied for any run of any agent.*

5. Results about commitment strategies

Because of our interest in reconciling commitment strategies and policy-based reconsideration (see section 2), we choose to verify if the blind (eq. 1) and single-minded (eq. 2) commitment strategies are valid in *AgentSpeak(L)*. Since policy-based reconsideration avoid seriously weighing desire-belief reasons for and against reconsideration ([3], p. 61), the open-minded commitment is not considered here, although it can be proved in a similar way to single-minded commitment. First, following the work on BDI_{CTL} [14], the no-infinite deferral axiom expressing that every intention is eventually dropped, is verified:

Proposition 1. *An* AgentSpeak(L) *agent satisfies the axiom of no-infinite deferral:*

$$\mathsf{INTEND}(\phi) \Rightarrow \mathsf{A}\Diamond(\neg\mathsf{INTEND}(\phi)). \tag{7}$$

Proof. Assume $K, c_0 \models \mathsf{INTEND}(\phi)$, then given the definition for INTEND (eq. 5), there is a plan $p \in C_I \cup C_E$ with head $+!\phi$ at c_0. The non-infinite deferral axiom expresses that for all runs K^0_Γ eventually this plan will be removed from C_I (active intentions) and C_E (suspended intentions). While p is being executed successfully, three runs are possible given the transition rules $ClrInt_X \in \Gamma$ (Appendix A): i) $ClrInt_3$ applies when the body of p is not empty, nothing is cleaned; ii) $ClrInt_2$ applies when the plan p, with an empty body, is at the top of an intention i, then p is dropped from i; $ClearInt_1$ applies when the intention i includes only the plan p with an empty body, the whole intention i is dropped. Given the finite nature of the plans (section 3.1), if everything goes right, the condition (iii) is eventually reached. If something goes wrong with p, a failure mechanism is activated by an event of the form $\langle -!\phi, i[p]\rangle$ resulting in p being dropped, so that condition (iii) is also eventually reached. □

Although the operational semantics of *AgentSpeak(L)* [2] only formalizes failures in finding relevant plans (see Appendix A, Rel_2), which discards suspended intentions in C_E, other forms of failure detection have been considered in the context of intentional learning [8, 9], e.g., when an action is not correctly executed; or when the expected effects of the plan are not believed by the agent after its execution.

Proposition 2. *An* AgentSpeak(L) *agent does not satisfy the axiom of blind commitment (eq. 1).*

Proof. Given that *AgentSpeak(L)* agents satisfy the no-infinite deferral property (eq. 7), and the semantics of weak until, the blind commitment axiom is reduced to (a similar reduction is used in Rao et al. [15]):

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow \mathsf{A}\Diamond\mathsf{BEL}(\phi) \tag{8}$$

Consider an initial configuration c_0 s.t. $ag_{bs} = \{\}$ and $ag_{ps} = \{+b(t_1) : \top \leftarrow p(t_2).\quad +!p(t_2) : \top \leftarrow +b(t_3).\}$. Suppose a belief update s.t. $ag_{bs} = \{b(t_1)\}$. An event is generated, so that $C_E = \{\langle +b(t_1), \top\rangle\}$. Following the transitions $SelEv_1$, Rel_1, and $ApplPl_1$ computes a new configuration where $C_I = \{[+b(t_1) : \top \leftarrow !p(t_2).]\}$ and $C_E = \{\}$. Proceeding with the transitions $SelAppl$, $ExtEv$, $SelInt_1$, $AchvGl$ computes a configuration c' where $C_E\langle +!p(t_2), +b(t_1) : \top \leftarrow \top\rangle$, $C_I = \{\}$. At this configuration, $K, c' \models \mathsf{DES}(p(t_2))$ (see eq. 6). Following the transitions $SelEv_1$, Rel_1, $AppPl_1$, $SelAppl$ computes a configuration c'' where $C_I = \{[+!p(t_2) : \top \leftarrow \ +b(t_3).]\}$ and $C_E = \{\}$. At this configuration, $K, c'' \models \mathsf{INTEND}(p(t_2))$ (see eq. 5). Proceeding with the transitions $IntEv$, $SelInt_1$, $AddBel$ gets $C_E = \langle +b(t_3), \top\rangle$, $ag_{bs} = \{b(t_1)\}$, $C_I = \{[+b(t_1) : \top \leftarrow \top \ddagger +!p(t_2) : \top \leftarrow \top]\}$ and $bs = \{b(t_1), b(t_3)\}$. The intention about $p(t_2)$ is maintained. Observe that the plan bodies in the intention are empty, so that $ClrInt_1$ and $ClrInt_2$ will discard the whole intention. At the next configuration c''', $K, c''' \models \neg\mathsf{INTEND}(p(t_2))$ and $K, c''' \models \neg\mathsf{BEL}(p(t_2))$. By counter-example the blind commitment axiom is not valid in *AgentSpeak(L)*. □

In fact, our agents do not satisfy the extended blind commitment axiom (eq. 1) neither, since the agent did not keep his intention about $p(t_2)$ until he believed it. This reasoning is similar to the demonstration of intention-belief incompleteness (AT2) for *AgentSpeak(L)* [1].

Proposition 3. AgentSpeak(L) *agents satisfy a limited form of the single-minded commitment:*

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$$

Proof. This case is similar to the no-infinite deferral demonstration. Assume that the agent $\mathsf{INTEND}(\mathsf{A}\Diamond\phi)$ at c_0, then there is a plan $p \in C_I \cup C_E$ with head $+!\phi$ at c_0. If there is a configuration $c_{k\geq 0}$ where $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ (the weak until definition has been adopted), then $K, x^{0,\ldots,k} \models \mathsf{INTEND}(\mathsf{A}\Diamond\phi)$. Following the no-infinite-deferral demonstration, in the plan execution failure cases the agent will eventually satisfy $\bigcirc\neg\mathsf{INTEND}(\phi)$ because of Rel_2, which means that for an event $\langle te, i[+!\phi : context \leftarrow body]\rangle$ there were not relevant plans to achieve ϕ so that it will be discarded, e.g., there is not a path of configurations where eventually ϕ would be achieved, so that it is rational to drop $\mathsf{INTEND}(\phi)$. The case of no-infinite deferral by successful execution of intentions covers the second condition of the weak until, when $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ does not occur. □

This is a limited form of single-minded commitement because $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ is not represented explicitly by the agents. In fact, the agent can not continue intending ϕ because there are no plans to do it and the intention fails. This is an instance of non-reflective reconsideration where preemptive abandon of intentions is not possible, i.e., the intention is dropped once it fails, not before. In order to obtain a full single-minded commitment enabling preemptive abandon of intentions, $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ needs to be represented explicitly by the agents, for instance as in policy-based reconsideration.

6. Conclusion

We have extended the methodology proposed by Bordini et al. [1] to reason about *AgentSpeak(L)* agents. Then we proved that the blind commitment strategy is not valid in this agent oriented programming language. On the other hand, a limited form of single-minded commitment is valid in *AgentSpeak(L)*. The main limitation for these agents with respect to commitment is the lack of an explicit representation for abandoning reasons. Guerra et al. [9, 10] have suggested that intentional learning provides a solution for this, enabling a policy-based reconsideration. In the intentional learning framework, when an intention fails, the agents learn by induction a first-order decision tree covering the successful executions of the plan that failed. In this way, the context of the plans that failed are redefined in terms of the branches of such trees, that lead to a success. Branches that lead to a failure can be used to define a policy for reconsideration.

Proving that *AgentSpeak(L)* agents are at least partially single-minded, enabled us to continue the research on a policy-based reconsideration based on intentional learning to accomplish full single-minded commitment. In recent experiments for a simple blocks world implemented in Jason [2], we found that policy-based reconsidering agents are much more tolerant to fast changes in the environment that the default partially single-minded agents, and event significantly more robust than intentional learning agents (reconsidering only the context of their plans). Interestingly, the degree of boldness and cautiousness for a given agent is something hard to define. It is well known [12] that in dynamic environments a very cautious agent performs better than a bold one; and inversely, in static environments boldness pays better. The relevance of learning intentionally is that the right degree of cautioness is learned by the agents, instead of being established once and forever by the programmers.

An extended *AgentSpeak(L)* operational semantics that deals with intentional learning, for both incremental and batch inductive methods, has been proposed by Guerra et al. [11]. It is inspired in the way *AgentSpeak(L)* is extended with speech acts: adding operational semantic rules that are implemented as a plan library. Using the techniques presented here, it is possible to arrive to a theory of commitment, reconsideration and learning, including rules like:

$$(\textbf{Abandon})\quad \frac{\mathcal{S}_{\mathcal{E}}(C_E) = \langle +abandon(i), \top\rangle \wedge ag_{bs} \models intending(i)}{\langle ag, C, M, T, SelEv\rangle \rightarrow \langle ag', C', M, T, SelEv\rangle}$$

s.t. $C'_E = C_E \backslash \{\langle +abandon(i), \top\rangle\}, ag_{bs} \not\models intending(i), C'_I = C_I \backslash i\rangle$

expressing that when an agent is intending i and an event with the trigger $+abandon(i)$ is generated by the policy learned by the agent, the intention is removed from C_I and the event is removed from C_E.

In this way we will be in position to experiment with these full singled-minded *AgentSpeak(L)* agents, and to prove if the expected properties introduced are valid or not using *CTL AgentSpeak(L)*. This illustrates the relevance of the specification language proposed in this paper, to bring *AgentSpeak(L)* closer to its philosophical foundation and to extend our computational theories of practical reasoning.

Acknowledgments

The first and third authors are supported by Conacyt CB-2007-1 (project 78910) and Promep CA-PIFI funding for this research. The second author is supported by Conacyt scholarship 214783.

A. AgentSpeak(L) operational semantics rules

The following *AgentSpeak(L)* operational semantics rules are relevant for the purposes of this paper, in particular for defining the Kripke structure supporting the semantics of temporal operators to reason about *AgentSpeak(L)* agents. They are adapted from Bordini et al. [2].

$$(\textbf{SelEv}_\textbf{1}) \quad \frac{\mathcal{S}_\mathcal{E}(C_E) = \langle te, i\rangle}{\langle ag, C, M, T, SelEv\rangle \rightarrow \langle ag, C', M, T', RelPl\rangle}$$

s.t. $C'_E = C_E \backslash \{\langle te, i\rangle\}, T'_\epsilon = \langle te, i\rangle$

$$(\textbf{Rel}_\textbf{1}) \quad \frac{T_\epsilon = \langle te, i\rangle, RelPlans(ag_{ps}, te) \neq \{\}}{\langle ag, C, M, T, RelPl\rangle \rightarrow \langle ag, C, M, T', AppPl\rangle}$$

s.t. $T'_R = RelPlans(ag_{ps}, te)$

$$(\textbf{Rel}_\textbf{2}) \quad \frac{RelPlans(ps, te) = \{\}}{\langle ag, C, M, T, RelPl\rangle \rightarrow \langle ag, C, M, T, SelEv\rangle}$$

$$(\textbf{Appl}_\textbf{1}) \quad \frac{ApplPlans(ag_{bs}, T_R) \neq \{\}}{\langle ag, C, M, T, ApplPl\rangle \rightarrow \langle ag, C, M, T', SelAppl\rangle}$$

s.t. $T'_{Ap} = AppPlans(ag_{bs}, T_R)$

$$(\textbf{SelAppl}) \quad \frac{\mathcal{S}_O(T_{Ap}) = (p, \theta)}{\langle ag, C, M, T, SelAppl\rangle \rightarrow \langle ag, C, M, T', AddIM\rangle}$$

s.t. $T'_\rho = (p, \theta)$

$$(\textbf{ExtEv}) \quad \frac{T_\epsilon = \langle te, \top\rangle, T_\rho = (p, \theta)}{\langle ag, C, M, T, AddIM\rangle \rightarrow \langle ag, C', M, T, SelInt\rangle}$$

s.t. $C'_I = C_I \cup \{[p\theta]\}$

$$(\textbf{SelInt}_\textbf{1}) \quad \frac{C_I \neq \{\}, \mathcal{S}_I(C_I) = i}{\langle ag, C, M, T, SelInt\rangle \rightarrow \langle ag, C, M, T', ExecInt\rangle}$$

s.t. $T'_\iota = i$

$$(\textbf{SelInt}_\textbf{2}) \quad \frac{C_I = \{\ \}}{\langle ag, C, M, T, SelInt\rangle \rightarrow \langle ag, C, M, T, ProcMsg\rangle}$$

$$(\textbf{AchvGl}) \quad \frac{T_\iota = i[head \leftarrow !at; h]}{\langle ag, C, M, T, ExecInt\rangle \rightarrow \langle ag, C', M, T, ProcMsg\rangle}$$

s.t. $C'_E = C_E \cup \{\langle +!at, T_\iota\rangle\}, C'_I = C_I \setminus \{T_\iota\}$

$$(\textbf{ClrInt}_1) \quad \frac{T_\iota = [head \leftarrow \top]}{\langle ag, C, M, T, ClrInt\rangle \rightarrow \langle ag, C', M, T, ProcMsg\rangle}$$

s.t. $C'_I = C_I \setminus \{T_\iota\}$

$$(\textbf{ClrInt}_2) \quad \frac{T_\iota = i[head \leftarrow \top]}{\langle ag, C, M, T, ClrInt\rangle \rightarrow \langle ag, C', M, T, ClrInt\rangle}$$

s.t. $C'_I = (C_I \setminus \{T_\iota\}) \cup \{k[(head' \leftarrow h)\theta]\}$ if $i = k[head' \leftarrow g; h]$ and $g\theta = TrEv(head)$

$$(\textbf{ClrInt}_3) \quad \frac{T_\iota \neq [head \leftarrow \top] \wedge T_\iota \neq i[head \leftarrow \top]}{\langle ag, C, M, T, ClrInt\rangle \rightarrow \langle ag, C, M, T, ProcMsg\rangle}$$

References

[1] R.H. Bordini and Á.F. Moreira. Proving BDI properties of agent-oriented programming languages. Annals of Mathematics and Artificial Intelligence 42 (2004) 197–226.

[2] R.H. Bordini, J.F. Hübner and M. Wooldridge. Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, England, 2007.

[3] M.E. Bratman. Intention, Plans, and Practical Reason. Harvard University Press, Cambridge, 1987.

[4] M.E. Bratman. Structures of Agency: Essays. Oxford University Press, New York, 2007.

[5] P. Cohen and H. Levesque. Intention is choice with commitment. Artificial Intelligence 42(3) (1990) 213–261.

[6] M. Dastani, M.B. van Riemsdijk and J.C. Meyer. A grounded specification language for agent programs, in: AAMAS '07. ACM, New York, NY, 2007, 1–8.

[7] A. Emerson. Temporal and modal logic, in: J. van Leeuwen (Ed.) Handbook of Theoretical Computer Science, Elsevier, Amsterdam, 1990, 996–1072.

[8] A. Guerra-Hernández, A. El-Fallah-Seghrouchni and H. Soldano. Learning in BDI Multi-agent Systems, in: J. Dix and J. Leite (Eds.), CLIMA IV. LNCS 3259, Springer, Heidelberg, 2004, 218–233.

[9] A. Guerra-Hernández, G. Ortíz-Hernández. Toward BDI sapient agents: Learning intentionally, in: R.V. Mayorga and L.I. Perlovsky (Eds.), Toward Artificial Sapience: Principles and Methods for Wise Systems, Springer, London, 2008, 77–91.

[10] A. Guerra-Hernández, J.M. Castro-Manzano, A. El-Fallah-Seghrouchni. Toward an AgentSpeak(L) theory of commitment and intentional learning, in: A. Gelbuch and E.F. Morales (Eds.), MICAI 2008. LNCS 5317, Springer-Verlag, Berlin Heidelberg, 2008, 848–858.

[11] A. Guerra-Hernández, G. Ortíz-Hernández and W.A. Luna-Ramírez. Jason smiles: Incremental BDI MAS learning, in: MICAI 2007 Special Session, IEEE Computer Society CPS, Los Alamitos, 2008, 61–70.

[12] D. Kinny and M.P. Georgeff. Commitment and effectiveness of situated agents, in: Proceedings of the twelfth international joint conference on artificial intelligence (IJCAI-91), Sydney, Australia, 1991.

[13] A.S. Rao and M.P. Georgeff. Modelling Rational Agents within a BDI-Architecture, in: M.N. Huhns and M.P. Singh (Eds.) Readings in Agents, Morgan Kaufmann, 1998, 317–328.

[14] A.S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language, in: W.V. de Velde and J.W. Perram (Eds.), MAAMAW. LNCS 1038, Springer, Berlin Heidelberg, 1996, 42–55.

[15] A.S. Rao and M.P. Georgeff. Decision procedures for BDI logics. Journal of Logic and Computation 8(3) (1998) 293–342.

[16] M.P. Singh. A critical examination of the Cohen-Levesque Theory of Intentions, in: Proceedings of the European Conference on Artificial Intelligence, 1992.

[17] M.P. Singh, A.S. Rao and M.P. Georgeff. Formal Methods in DAI: Logic-Based Representation and Reasoning, in: G. Weiss (Ed.), Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge, 1999, 331–376.

[18] M. Wooldridge. Reasoning about Rational Agents. MIT Press, Cambridge, 2000.