Communicating Rational Agents: Semantics and Verification

Nils Bulling\textsuperscript{1} and Koen V. Hindriks\textsuperscript{2}

\textsuperscript{1}Department of Informatics
Clausthal University of Technology, Germany
bulling@in.tu-clausthal.de

\textsuperscript{2}Faculty Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, The Netherlands
k.v.hindriks@tudelft.nl

Abstract. We present a computational semantics of communicative actions for rational agent programming languages. Three indicators are used to differentiate declarative, interrogative and imperative messages which replace the usual labels to identify speech acts. We introduce a multi-agent verification logic based on the computational semantics that facilitates reasoning about communicative actions. Subsequently, this multi-agent logic is embedded into a more expressive modal logic over a standard run-based semantics. We formally relate both logics, prove expressivity results, and argue why it is an advantage to have a more expressive standard modal logic at hand.

1 Introduction

We introduce a computational semantics for communicative actions based on mental models. A shift is made from the traditional speech-act based labels to the exchange of messages differentiated only by grammatical markers. Three markers corresponding to the three sentence types declarative, interrogative, and imperative, which are also distinguished in natural language grammar, are introduced. We believe the notion of a speech act is best used descriptively to characterize message exchanges between agents.

The semantics is designed such that it is particularly easy to integrate it into agent programming languages that facilitate programming with mental models (e.g. [9, 1, 4] nbu:Your PROMAS Paper\textsuperscript{?}, where a mental model consists of the declarative beliefs and goals of an agent. To this end, a transition semantics is introduced that provides a recipe for implementing communication between such agents. We then continue to show that this semantics can be embedded into a modal semantics. This result can be seen as a conservative extension of the single-agent result presented in [9] to the multi-agent setting introduced here; however, in this paper we left out some operators whose addition is straightforward. The modal logic can be used to reason about communicating agents in a more expressive (and external) way than the verification language allows.
for. Moreover, standard modal logic techniques and results for modal logics, e.g. modal checking, can be applied.

Section 2 introduces the basic multi-agent model presupposed by both the transition as well as the modal semantics. Section 3 introduces the essentials of an agent programming language on top of which Section 4 adds a computational semantics for communicative actions. We also introduce a logic based on the transition semantics. Section 5 embeds this semantics into a modal semantics. The modal logic is used in Section ?? to characterize message exchanges as speech acts. Section 6 concludes the paper.

2 Preliminaries: The Multi-Agent Model

The multi-agent model assumes a fixed number of agents with associated agent names $\textit{Agt} = \{a_1, \ldots, a_n\}$. This assumption is not essential here but simplifies the technical presentation. A global state $g$ of a multi-agent system ($\textit{Mas}$) is a tuple $(l_{a_1}, \ldots, l_{a_n}, l_e)$ with $l_{a_i}$ the local state of agent $a_i$ and $l_e$ the state of the environment. We use $g_a$ to denote the local state of agent $a$. The non-empty set $G = L_{a_1} \times \ldots \times L_{a_n} \times L_e$ represents all (global) states.

In each state an agent $a$ may perform an action, drawn from a set of actions $\textit{Act}_a$. We use $\alpha_i$ to denote actions and assume that $\textit{Act}_a$ contains an action $\Lambda$ which corresponds to agent $a$ performing no action. Without loss of generality, we assume that $\textit{Act}_a \cap \textit{Act}_b = \emptyset$ when $a \neq b$. $\textit{Act}$ denotes the union of the action sets of all agents. As we abstract from the details of action selection here, we associate mappings $P_a : L_a \rightarrow \mathcal{P}(\textit{Act}_a)$ called programs with each agent that map a local state to a non-empty set of actions from which the agent may nondeterministically select an action to perform. An agent may then be defined as a pair $(a, P_a)$ with $a \in \textit{Agt}$ and $P_a$ a program. Usually, we identify an agent with its name.

The effects of performing an action are represented by a transition function $\tau : G \times \textit{Act} \rightarrow G$. Actions are assumed to update only the local state of the agent performing it. That is, $\tau(g, \alpha)_a = g_a$ whenever $\alpha \notin \textit{Act}_a$. An exception to this rule will be made below for communicative actions.

The behavior of a multi-agent system is given by a run $r$ which is a mapping $\mathbb{N} \rightarrow G \times \textit{Act}$. $r_1(i)$ (resp. $r_2(i)$) is used to denote the projection of $r(i)$ onto the first (resp. second) component of $r(i)$. We thus use an interleaving semantics to model the execution of a MAS. This means that for every $i$ we have $r_2(i) \in P_a(g_a)$ for some $a \in \textit{Agt}$ where $g_a = (r_1(i))_a$ and $r_1(i + 1) = \tau(r(i))$. Additional constraints such as fairness may be added but are not studied in this paper.

Definition 1 (Mas Model). A multi-agent system model $\mathcal{R}$, system for short, is defined as a set of runs.

3 Programming with Mental Models

Rational agents are programs that derive their choice of action from their beliefs and goals. Agent programming language provides a framework for programming
with *mental models* that consist of an agent’s beliefs and goals. Whereas in the single agent setting a mental model consists of the agent’s own beliefs and goals only, in the multi-agent setting, that we consider here, we introduce the notion of a mental state that consists of mental models of other agents as well. The idea is that these mental models are used to (partially) reconstruct the beliefs and goals of another agent, given, for example, the messages received from that agent or observations of other actions performed by that agent. Such mental models may also be used in an agent program, for example, to include and represent beliefs that are considered common knowledge. In Section 4 we show how messages received from another agent may be used to construct a mental model of that agent.

We reserve the label *mental state* for composed entities consisting of multiple mental models, one for each agent in the MAS, and the label *mental model* for pairs of beliefs and goals associated with a particular agent. A mental state of an agent thus is a mapping from agent names to mental models. The beliefs and goals of an agent are declarative sentences which are represented in some underlying knowledge representation technology. Here we assume a propositional language $\mathcal{L}_{PL}$ built from a set of propositional atoms $\text{Atom}$ and the usual boolean connectives. $\models_{PL}$ denotes the usual consequence relation associated with $\mathcal{L}_{PL}$, and the special symbol $\bot \in \mathcal{L}_{PL}$ denotes the false proposition. As is common [4, 9], some additional rationality constraints are imposed on a mental model.

**Definition 2 (Mental Models and Mental States).** A mental model is a pair $\langle \Sigma, \Gamma \rangle$ with $\Sigma \subseteq \mathcal{L}_{PL}$ a belief and $\Gamma \subseteq \mathcal{L}_{PL}$ a goal base which satisfy the following rationality constraints:

- The belief base is consistent: $\Sigma \not\models_{PL} \bot$;
- Individual goals are consistent: $\forall \gamma \in \Gamma : \gamma \not\models_{PL} \bot$;
- Goals are not believed to be achieved: $\forall \gamma \in \Gamma : \Sigma \not\models_{PL} \gamma$.

A mental state is a mapping $m$ from $\text{Agt}$ to mental models, i.e. $m(a) = \langle \Sigma, \Gamma \rangle$ is a mental model for each $a \in \text{Agt}$. The set of all mental states is denoted by $\text{MS}(\text{Agt})$.

The intuition is that a mental state $m_a$ encodes $a$’s beliefs about $b$’s beliefs and goals by mapping agent name $b$ to a mental model $m_a(b) = \langle \Sigma, \Gamma \rangle$ where $\Sigma$ encodes $b$’s beliefs and $\Gamma$ encodes $b$’s goals. Agent $a$’s beliefs about agent $b$ do not have to correspond with the actual mental state of agent $b$, i.e. it may be the case that $m_a(b) \neq m_b(b)$ where $m_b$ denotes the mental state of agent $b$. Agent $a$’s private belief and goal bases are accessed by $m_a(a)$. A mental state thus allows for *second-order beliefs* but not higher-order beliefs as we want to ensure the model is both tractable as well as a useful extension of current agent programming languages. To ensure this we need to make a trade-off between expressivity and computational complexity here. As a consequence, in the model proposed, we cannot have, for example, that an agent believes that another agent either believes $\phi$ or believes $\neg \phi$.

Mental states are concrete instantiations of the local states of Section 2 and we get $G = \text{MS}_{a_1} \times \ldots \times \text{MS}_{a_n}$ where $\text{MS}_{a_i}$ denotes the set of mental states for
agent $a_i$. Agents need to be able to inspect their mental state and the different mental models part of it. In order to do so we introduce belief ($B_a^\phi$) and goal modalities ($G_a^\phi$) which are annotated with agent names in order to refer to the beliefs and goals, respectively, in the mental model associated with agent $a$.

**Definition 3 (Mental State Conditions).** The language of mental state conditions over $\mathcal{Agt}$, $\mathcal{L}_{MS}(\mathcal{Agt})$, is defined by: $\psi ::= B_a^\phi \mid G_a^\phi \mid \neg \psi \mid \psi \land \psi$ where $a \in \mathcal{Agt}$ and $\phi \in \mathcal{L}_{PL}$.

The semantics of mental state conditions is defined relative to a mental state. The truth of $B_b^\phi$ in a mental state $m_a$ of agent $a$ denotes that agent $a$ believes that agent $b$ believes $\phi$, and, similarly, $G_b^\phi$ denotes that $a$ believes $b$ has goal $\phi$. That agent $a$ has these beliefs is left implicit in an agent program as this program provides the context in which to evaluate a mental state condition. In case $a = b$ and $B_a^\phi$ (resp. $G_a^\phi$) is true we simply say that agent $a$ believes (resp. has goal) $\phi$.

**Definition 4 (Semantics of $\mathcal{L}_{MS}$).** Let $m$ be a mental state and $m(a) = \langle \Sigma_a, \Gamma_a \rangle$. The semantics of $\mathcal{L}_{MS}$ is defined by:

- $m \models_{MS} B_a^\phi$ iff $\Sigma_a \models_{PL} \phi$
- $m \models_{MS} G_a^\phi$ iff $\exists \gamma \in \Gamma_a$ such that $\gamma \models_{PL} \phi$
- $m \models_{MS} \neg \psi$ iff $m \not\models_{MS} \psi$
- $m \models_{MS} \psi \land \psi'$ iff $m \models_{MS} \psi$ and $m \models_{MS} \psi'$

The behavior of an agent is determined by its mental state. The transition functions $\tau$ of Section 2 therefore will also be called mental state transformers; a corresponding run is also called a(n) (agent) trace. As a mental state transformer has type $MS \times Act \rightarrow MS$, $\tau(g, \alpha)_a(b)$ must satisfy the constraints from Definition 2 for any $a, b \in \mathcal{Agt}$. Here, $\tau(g, \alpha)$ denotes a global state, $\tau(g, \alpha)_a$ denotes the mental state of agent $a$, and $\tau(g, \alpha)_a(b)$ denotes the mental model agent $a$ associates with agent $b$, a notation we will often use below.

### 4 Communicating Agents

Actions have been defined abstractly as mental state transformers that affect an agent’s own mental state. The communicative actions that we introduce here affect the mental state of the receiving agent and may be viewed as complementary to actions that act upon the environment and are useful in a multi-agent context to affect other agents. A communicative action is of the form $send(a, b, msg) \in Act_a$ where $msg$ denotes a message that is being sent by agent $a$ to $b$. This differs from the traditional approach based on speech act theory, where such actions are of the form $send(a, b, label, msg)$ where $label$ identifies the type of speech act performed (e.g. inform, query, etc.; see e.g. [10, 8, 11]) and $msg$ denotes the message content. One of the main problems, extensively discussed in the literature, concerns the use of speech-act
identifying labels as part of the message sent. It is, for example, not possible for
the receiving agent to verify that the label used to identify the speech act per-
formed corresponds to the act that is actually performed by the sending agent
(cf. [13]).

To avoid such problems, we do not use speech-act labels but instead use
indicators to identify the sentence type. Three indicators are introduced that
intuitively correspond with the sentence types most often used in natural lan-
guage: • for declarative, ? for interrogative, and ! for imperative sentences. The
set $\mathcal{M}_{\text{sg}}$ of messages consists of all messages of the form $\bullet \phi$, $\? \phi$, or $\! \phi$ where
$\phi \in \mathcal{L}_{PL}$.

Semantically, a shift from the focus on the sender to the hearer is proposed.
It is natural that upon receiving a message an agent will update its mental
state. The main question then is how? Taking as our example a message $\bullet \phi$ with
declarative indicator, different options are available: The receiving agent (i) in-
corporates $\phi$ into its own beliefs, (ii) comes to believe that the sender believes
$\phi$, or (iii) comes to believe the sender intended the receiving agent to believe
$\phi$. Other, less useful options, are ignored here. We take a definite engineering
stance here as our main interest is in providing useful communication primitives
for programming multi-agent systems which also correspond with basic common
sense intuitions. Taking as our starting point, we argue that: option (i) is too
strong as it introduces the assumption that the sender always convinces the
receiver, and (iii) is too weak as it is no longer very clear what use the commu-
nication has and quite complex reasoning patterns would be needed to conclude
something useful from such indirect information about the sender’s mental state.
This leaves option (ii) which, although it may not always be safe, instead only
makes the assumption that the sender believes what it says and arguably takes
a message at its face value. This choice also seems reasonable as a default
way to handle messages from the pragmatic perspective. For example, the Maxim of
Quality, which instructs to not say what you believe to be false, or for which you
lack adequate evidence, may be viewed as supporting the view that taking mes-
sages at face value as option (ii) would provide an adequate default to interpret
a message.

We continue to formalize the informal discussion above by extending the
mental state transformer $\tau$ such that it can be applied to $\text{send}(a, b, i \phi)$ actions
as well ($i \in \{?, \bullet, \!\}$). In order to simplify the presentation, we assume that
whenever a message is sent it is immediately received and processed by the
recipient.

**Definition 5 (Message-Passing Mst).** A message-passing mental state trans-
former $\tau : G \times \text{Act} \to G$ satisfies:

- • if $b \neq a$, $\alpha \in \text{Act}_a$, $\alpha \neq \text{send}(a, b, m)$, then $\tau(g, \alpha)_b = g_b$
- • if $\alpha = \text{send}(a, b, m)$, then (i) $\tau(g, \alpha)_i = g_i$, $\forall i \in \text{Agt} \setminus \{b\}$,
  (ii) $\tau(g, \alpha)_b(i) = g_b(i)$ $\forall i \in \text{Agt} \setminus \{a\}$, and
Communicating a message \( m \) thus modifies the mental model \( \langle \Sigma_a, \Gamma_a \rangle \) of the 
sender \( a \) maintained by receiver \( b \) as follows:

1. When \( m \) is a declarative, \( \phi \) is added to the belief base \( \Sigma_a \) to represent the 
belief of agent \( b \) that \( a \) believes \( \phi \). The addition of \( \phi \) to \( \Sigma_a \) is modelled by 
\( \oplus \), where we minimally assume that \( \Sigma_a \oplus \phi \models_{PL} \phi \) when \( \phi \) is consistent. 
Moreover, we have to remove goals which are satisfied wrt. the updated 
belief base in order to meet the third rationality constraint from Def. 2.

2. When \( m \) is an interrogative, \( \phi \) is removed from the belief base \( \Sigma_a \) to represent 
the belief that agent \( a \) does not believe \( \phi \) when it asks a question \( ?\phi \). The 
removal of \( \phi \) from \( \Sigma_a \) is modelled by \( \ominus \), where we minimally assume that 
\( \Sigma_a \ominus \phi \not\models_{PL} \phi \) when \( \phi \) is not a tautology.

3. When \( m \) is an imperative, and \( \phi \) is not a tautology, then it is added to the goal 
base \( \Gamma_a \) to represent the belief that agent \( a \) wants \( \phi \) when it communicates 
\( !\phi \) and \( \phi \) is removed from the belief base \( \Sigma_a \) (cf. Definition 2).

**The Verification Language \( \mathcal{L}_V \).** The temporal language \( \mathcal{L}_V \) to reason about 
communicating agents is an extension of the verification logic introduced in \cite{9}.

In the logic we need to refer to the beliefs an agent has about the beliefs and 
goals of another agent, and therefore, a superscript is added to the belief and 
goal modalities where \( B_b^a \phi \) represents that \( a \) believes that \( b \) believes \( \phi \), and, 
similarly, \( G_b^a \phi \) represents that \( a \) believes that \( b \) has goal \( \phi \).

**Definition 6 (Verification Language \( \mathcal{L}_V \)).** \( \mathcal{L}_V \) denotes the set of formulae \( \chi \) defined by the following grammar:

\[
\chi ::= B_b^a \phi \mid G_b^a \phi \mid \neg \chi \mid \chi \land \chi \mid \chi U \chi \mid X \chi \mid \text{done}_a(\alpha)
\]

where \( \phi \in \mathcal{L}_{PL} \), \( \alpha \in \text{Act}_a \) and \( a,b \in \text{Agt} \). We also write \( B_a^\emptyset \) for \( B_a^a \) and \( G_a^\emptyset \) for \( G_a^a \).

A trace generated by several goal agents and a message passing mental state 
transformer serves as a model for \( \mathcal{L}_V \). Given such a trace and a time point, the 
semantics of \( \mathcal{L}_V \)-formulae is defined in a straightforward way.

\footnote{Apart from the global modality \([\alpha]\) referring to action executions which is left out 
in this paper.}
Definition 7 (Semantics of $\mathcal{L}_V$). The semantics of $\mathcal{L}_V$-formulae is defined relative to a trace $t$ and a time point $i \in \mathbb{N}$:

\[
\begin{align*}
t, i \models_V B_a \phi & \iff g_a \models_{MS} B^b \phi \text{ where } g = t_1(i) \\
t, i \models_V G_a \phi & \iff g_a \models_{MS} G^b \phi \text{ where } g = t_1(i) \\
t, i \models_V \neg \chi & \iff t, i \not\models_V \chi \\
t, i \models_V \chi \land \chi' & \iff t, i \models_V \chi \text{ and } t, i \models_V \chi' \\
t, i \models_V X \chi & \iff t, i + 1 \models_V \chi \\
t, i \models_V \chi U \chi' & \iff \exists j \geq i : t, j \models_V \chi' \text{ and } \\
& \forall k : i \leq k < j \Rightarrow t, k \models_V \chi \\
t, i \models_V done_a(\alpha) & \iff i > 0 \text{ and } t_2(i - 1) = \alpha
\end{align*}
\]

5 Embedding $\mathcal{L}_V$ in the Modal Logic $\mathcal{L}_M$

In this section we introduce the modal logic $\mathcal{L}_M$ which is used to reason about runs. Then, we relate the verification logic $\mathcal{L}_V$ and its semantics to the modal logic $\mathcal{L}_M$ and present expressiveness results.

5.1 $\mathcal{L}_M$: Syntax and Semantics

The language $\mathcal{L}_M$ is built from atoms $p \in \text{Atom}$ and the temporal constructs $\bigcirc \varphi$ for $\varphi$ holds in the next state, $\varphi \mathcal{U} \psi$ for $\varphi$ holds until $\psi$ holds, belief operators $B_a \varphi$ for $a \in \text{Agt}$ believes $\varphi$, goal operators $G_a \varphi$ for $a$ has goal $\varphi$, and $\text{Done}_a(\alpha)$ for $a$ has performed $\alpha \in \text{Act}$. The semantics, however, is already ready to handle them.

Definition 8 ($\mathcal{L}_M$). The language $\mathcal{L}_M$ is defined by:

\[
\varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid B_a \varphi \mid G_a \varphi \mid \bigcirc \varphi \mid \varphi \mathcal{U} \psi \mid \text{Done}_a(\alpha)
\]

The behavior of a MAS is modelled by a set of runs (cf. Section 2); in addition, an $\mathcal{L}_M$-model consists of the usual belief and goal accessibility relations for each agent, and a valuation which labels states with the facts true in it.

Definition 9 ($\mathcal{L}_M$-Model). An $\mathcal{L}_M$-model $\mathfrak{M}$ is a tuple $\langle \mathcal{R}, \{B_a \mid a \in \text{Agt}\}, \{G_a \mid a \in \text{Agt}\}, V \rangle$ where $\mathcal{R}$ is a set of runs over states $G$ and actions $\text{Act}$, $B_a, G_a \subseteq \mathcal{R} \times \mathbb{N} \times \mathcal{R} \times \mathbb{N}$ are serial belief and goal accessibility relations, respectively, and $V : \mathcal{R} \times \mathbb{N} \rightarrow \mathcal{P}(\text{Atom})$ is a valuation function, which assigns to each point the propositional atoms true in it.

Formulae are interpreted over $\mathcal{L}_M$-models in the standard way (see e.g. [7]). We use $\mathfrak{M}, r, i \models \varphi$ to denote that $\varphi$ is satisfied on $r$ at time $i$ in model $\mathfrak{M}$.

\[\text{Due to space restrictions, some useful other operators have been left out, see e.g. [6, 9].}\]
Definition 10 (Semantics of $\mathcal{L}_M$). The semantics of $\mathcal{L}_M$-formulae over an $\mathcal{L}_M$-model $\mathcal{M}$, a run $r \in \mathcal{R}_\mathcal{M}$, and a time point $i \in \mathbb{N}$ is defined as follows:

- $\mathcal{M}, r, i \Vdash p$ if and only if $p \in V(r, i)$
- $\mathcal{M}, r, i \Vdash \neg \varphi$ if and only if $\mathcal{M}, r, i \not\Vdash \varphi$
- $\mathcal{M}, r, i \Vdash \varphi \land \psi$ if and only if $\mathcal{M}, r, i \Vdash \varphi$ and $\mathcal{M}, r, i \Vdash \psi$
- $\mathcal{M}, r, i \Vdash B_\alpha \varphi$ if and only if $\forall (r', i') \in B_\alpha(r, i) : \mathcal{M}, r', i' \Vdash \varphi$
- $\mathcal{M}, r, i \Vdash G_\alpha \varphi$ if and only if $\forall (r', i') \in G_\alpha(r, i) : \mathcal{M}, r', i' \Vdash \varphi$
- $\mathcal{M}, r, i \Vdash \Diamond \varphi$ if and only if $\mathcal{M}, r, i + 1 \Vdash \varphi$
- $\mathcal{M}, r, i \Vdash \varphi \land \psi$ if and only if $\exists j : j \geq i$ and $\mathcal{M}, r, j \Vdash \psi$ s.t. $\forall k : i \leq k < j \Rightarrow \mathcal{M}, r, k \Vdash \varphi$
- $\mathcal{M}, r, i \Vdash \text{Done}_a(\alpha)$ if $i > 0$ and $r_2(i - 1) = \alpha \in \text{Act}_a$

We define $X_a(r, i) = \{(r', i') \mid X_a(r, i, r', i')\}$ for $X \in \{B, G\}$ and, as usual, abbreviate $B_\alpha \varphi \land \varphi$ as $K_\alpha \varphi$.

5.2 Equivalence and Correspondence Results

We formally relate the logics $\mathcal{L}_V$ and $\mathcal{L}_M$ by embedding $\mathcal{L}_V$ into $\mathcal{L}_M$. We do so by introducing a translation $tr$ from $\mathcal{L}_V$-formulae to $\mathcal{L}_M$-formulae and showing that this translation preserves truth. This shows that $\mathcal{L}_M$ can be used to reason about communicating agents instead of the non-standard $\mathcal{L}_V$, which is not only useful as standard techniques for modal logic can be applied but also because it allows us to study expressiveness of $\mathcal{L}_V$ compared to $\mathcal{L}_M$.

First, we define the syntactic translation of formulae.

Definition 11 (Translation $tr : \mathcal{L}_V \rightarrow \mathcal{L}_M$).

- $tr(B^b_\alpha \varphi) = \begin{cases} B_\alpha B_\beta \varphi & \text{if } a \neq b \\ B_\alpha \varphi & \text{if } a = b \end{cases}$
- $tr(G^b_\alpha \varphi) = \begin{cases} B_\alpha G_\beta \boxdot \varphi & \text{if } a \neq b \\ G_\alpha \boxdot \varphi & \text{if } a = b \end{cases}$
- $tr(\neg \varphi) = \neg tr(\varphi)$
- $tr(\varphi \land \psi) = tr(\varphi) \land tr(\psi)$
- $tr(X \varphi) = \circ tr(\varphi)$
- $tr(\varphi U \phi) = tr(\varphi) U tr(\psi)$
- $tr(\text{Done}_a(\alpha)) = \text{Done}_a(\alpha)$

Our first result shows that $\mathcal{L}_M$ and its models are at least as expressive as $\mathcal{L}_V$ over traces; i.e., the modal logic can be used to reason about traces.

Theorem 1. Let $t$ be a trace. Then there is an $\mathcal{L}_M$-model $\mathcal{M} = \langle \mathcal{R}, \{B_a \mid a \in \text{Act}\}, \{G_a \mid a \in \text{Act}\}, V \rangle$ and a run $r^i \in \mathcal{R}$ such that for all $\varphi \in \mathcal{L}_V$ and $i \in \mathbb{N}$ we have: $t, i \Vdash \varphi$ if and only if $\mathcal{M}, r^i, i \Vdash tr(\varphi)$.

Proof. (Sketch) Let $\Pi(T) := \{\pi \subseteq \text{Atom} \mid \pi \models_{\mathcal{PL}} T\}$ denote the set of all valuations that make $T \subseteq \mathcal{L}_{\mathcal{PL}}$ true. The set of runs $\mathcal{R}$ is defined over states
G := L_a \times \ldots \times L_a \times L_c. We set L_e := \mathcal{P}(\text{Atom}) and L_a := MS_a for a \in \mathcal{A}gt. The run r^t corresponding to trace t is simply defined by: \( r_2(i) := t_2(i), (r_1(i))_a := \emptyset \), and \((r_1(i))_a := (t_1(i))_a \) for all \( i \in \mathbb{N}, a \in \mathcal{A}gt \).

To obtain the belief and goal accessibility relations we define a set \( \mathcal{R}^{b} \) of all “belief-reachable” runs. That is, \( \mathcal{R}^{b} := \{ r \mid \exists k : (r_1(k) \neq (\emptyset, m_\emptyset, \ldots, m_\emptyset)), \forall l : r_2(l) = \epsilon \} \) where \( \epsilon \not\in \text{Act} \) and \( m_\emptyset \) maps all agent names to \( (\emptyset, \emptyset) \). The belief accessibility relation for agent \( a \) is now defined as follows: \( \mathcal{B}_a(r, i, r', i') \) iff (i) \( r' \in \mathcal{R}^{b} \), (ii) \( i = i' \), (iii) \( r_1(i) \in \Pi(\Sigma_a) \) where \((r_1(i))_a(a) = (\Sigma_a, \Gamma_a) \), and (iv) \( \forall b \in \mathcal{A}gt ((r_1'(i))_b = (r_1(i))_a) \). That is, a run is \( \mathcal{B}_a \)-reachable if each local state \( l_b \) represents \( a \)’s mental state (given in the run it is in relation with); and the environment state is used to encode \( a \)’s beliefs about the world.

Let \( \mathcal{R}(\{\gamma_1, \gamma_2, \ldots \}, i) := \{ r \mid \forall j : (r_1(i + j))_a \in \Pi(\{\gamma_1\}), \forall k \in \mathbb{N}a \in \mathcal{A}gt : (r_1(k))_a = m_\emptyset \}. \) Now we define the goal accessibility relation as follows: \( \mathcal{G}_a(r, i, r', i') \) iff (i) \( i = i' \) and (ii) \( r' \in \mathcal{R}(i, i) \) where \((r_1(i))_a(a) = (\Sigma_a, \Gamma_a) \). That is, each “goal run” contains one valuation for each goal in \( \Gamma_a \) at some future time point.

Finally, we set \( \mathcal{R} := \{ r^t \} \cup \mathcal{R}^{b} \cup \mathcal{R}^{g} \) where \( \mathcal{R}^{g} := \bigcup_{X \subseteq \mathcal{L}_{\text{PL}}, i \in \mathbb{N}} \mathcal{R}(X, i) \) and \( V(r, i) := \{ p \in \text{Atom} \mid (r_1(i))_a \models_{\text{PL}} p \} \). That \( t, i \models \varphi \) iff \( \mathcal{M}, r^t, i \models \text{tr}(\varphi) \) is straightforwardly shown by structural induction on \( \varphi \).

To obtain a correspondence result in the other direction, it is clear we need to impose some constraints on \( \mathcal{L}_{\text{MS}} \)-models to ensure they model mental states and meet the conditions of Definition 2 and model communicative actions as in Definition 5. Firstly, consider the rationality conditions of Definition 2. The first two consistency conditions are satisfied because of the seriality of the belief and goal relations. An additional postulate is introduced to match the third condition. It is helpful to first define some notation: \( \mathcal{B}^b_a(r, i) := (\mathcal{B}_b \circ \mathcal{B}_a)(r, i) = \{ (r', i') \mid \exists (r'', i'') \in \mathcal{B}_a(r, i) : (r', i') \in \mathcal{B}_b(r'', i'') \} \). i.e. \( \mathcal{B}^b_a(r, i) \) contains all points which are \( \mathcal{B}_b \)-reachable from some point which is \( \mathcal{B}_a \)-reachable from \( (r, i) \); \( \mathcal{G}^b_a := \mathcal{G}_b \circ \mathcal{B}_a \) is defined analogously. To match the third condition, we now introduce the following postulate:

\[(\text{R1}) \quad \forall a, b \in \mathcal{A}gt : \mathcal{G}^b_a(r, i) \subseteq \langle \varphi \rangle_{\mathcal{M}, r}, i \Rightarrow \mathcal{B}^b_a(r, i) \subseteq [\varphi]_{\mathcal{M}}, \rangle \]

where \([\varphi]_{\mathcal{M}, r} := \{ (r, i) \mid \mathcal{M}, r, i \models \varphi \} \), the denotation of \( \varphi \), consists of the points that satisfy \( \varphi \). The subscript \( \mathcal{M} \) is omitted if clear from context.

In order to be able to match the communication semantics of Definition 5, two additional postulates are required. Let \( r \) be a run and \( X \in \{ B, G \} \). The following condition says that only the beliefs and goals of an action executing agent may change provided it is not a send action:

\[(\text{R2}) \quad \text{If send}(\cdot, \cdot, \text{msg}) \neq r_2(i) \in \text{Act}_a \text{ then for all } c, d \in \mathcal{A}gt, c \neq a : X_c(r, i) = X_c(r, i + 1) \text{ and } X^d_c(r, i) = X^d_c(r, i + 1) \]

Our last postulate handles the case when a message is sent: Only the mental state of the agent who receives the message is allowed to change in a prescribed way.
(R3) If \( r_2(i) = \text{send}(a, b, \text{msg}) \) then for all \( c, d \in \text{Agt} \): \( X_c(r, i) = X_c(r, i + 1) \) and \( X_d^a(r, i) = X_d^a(r, i + 1) \) except if:
\[ \text{msg} = \bullet \varphi \quad \text{and} \quad \varphi \text{ consistent then } B_0^a(r, i + 1) \subseteq [\varphi]; \]
\[ \text{msg} = \lnot \varphi \quad \text{and} \quad \varphi \text{ no tautology then } B_0^a(r, i + 1) \not\subseteq [\varphi]; \]
\[ \text{msg} = \lnot \varphi \quad \text{and} \quad \varphi \text{ no tautology then } G_0^a(r, i + 1) \not\subseteq G_0^a(r, i) \cap [\lnot \varphi]. \]

Note that in the case of \( \bullet \varphi \), (R1) ensures that \( \varphi \) is not a goal. Finally, we call a run \textit{trace-consistent} if it satisfies conditions (R1), (R2), and (R3) and an \( \mathcal{L}_M \)-model is said to be \textit{trace-consistent} if it contains at least one trace-consistent run.

**Theorem 2.** Let \( \mathfrak{M} \) be a trace-consistent \( \mathcal{L}_M \)-model. For each trace-consistent run \( r, \) all \( \varphi \in \mathcal{L}_V \), and \( i \in \mathbb{N} \): \( \mathfrak{M}, r, i \models \varphi \) iff.

**Proof.** (Sketch) Let \( \mathfrak{M} = (\mathcal{R}, \{B_a \mid a \in \text{Agt}\}, \{G_a \mid a \in \text{Agt}\}, V) \) and \( r \in \mathcal{R} \) a trace-consistent run. The trace \( t \) is defined from \( r \) as follows. For all \( i \in \mathbb{N} \) we set (1) \( t_2(i) := t_2(i) \) and (2) for all \( a, b \in \text{Agt} \) we set \( (t_1(i))_a(b) := (\Sigma_a^b, \Gamma_a^b) \) where for \( a \neq b \): \( \Sigma_a^b := \{ \phi \in \mathcal{L}_{PL} \mid \mathfrak{M}, r, i \models B_a B_b \phi \} \), \( \Gamma_a^b := \{ \phi \in \mathcal{L}_{PL} \mid \mathfrak{M}, r, i \models G_a \phi \} \); and for \( a = b \): \( \Sigma_a^b := \{ \phi \in \mathcal{L}_{PL} \mid \mathfrak{M}, r, i \models B_a \phi \} \), \( \Gamma_a^b := \{ \phi \in \mathcal{L}_{PL} \mid \mathfrak{M}, r, i \models G_a \phi \} \). The following claims complete the proof.

**Claim 1:** \( t \) is a trace. Proof: That each element of \( t_1(i) \) is consistent is ensured by the seriality of the belief relations. Assume some \( \Sigma_a^b \) is inconsistent. Then there are \( \varphi, \psi \in \Sigma_a \) such that \( \varphi \land \psi \) is inconsistent. Then, \( \mathfrak{M}, r, i \models B_a (\varphi \land \psi) \) and thus also \( \mathfrak{M}, r, i \models B_a \bot \) which contradicts the seriality of \( B_a \). Following the same reasoning, we get that each goal contained in any goal base must be consistent. Finally, that goals are not believed is ensured by postulate (R1).

Assume that \( \mathfrak{M}, r, i \models G_a \Diamond \gamma \land B_a \gamma \), then we would have \( B_a(r, i) \subseteq [\gamma] \) and \( G_a(r, i) \subseteq [\Diamond \gamma] \), which is a contradiction. It remains to show that \( t \) satisfies the conditions imposed by Def. 5. For actions other than communicative actions postulate (R2) guarantees that only the mental state of the agent that performs it is affected. For a communicative act, assume that \( t_2(i - 1) = \text{send}(a, b, \text{msg}) \) and let \( (t_1(i - 1))_b(a) = (\Sigma_a^b, \Gamma_a^b) \) and \( (t_1(i))_b(a) = (\Delta_a^b, \Gamma'_a^b) \).

**Case** \( \text{msg} = \bullet \psi \): For consistent \( \psi \), we show that \( \Sigma'_a \models_{PL} \psi \). Assume this is not the case, i.e. \( \mathfrak{M}, r, i \not\models B_b B_a \psi \). This is equivalent to \( \neg(\forall (r', i') \in B_b(r, i) : (\exists (r'', i'') \in B_a(r', i') : (\mathfrak{M}, r'', i'' \models \psi)) \) which in turn is equivalent to \( \exists (r'', i'') \in B_b(r, i) : (\mathfrak{M}, r'', i'' \not\models \psi) \). But this contradicts (R3): \( B_b(r, i) \subseteq [\psi] \). Moreover, it cannot be the case that there is a \( \gamma \in \Gamma_a^b \) such that \( \Sigma'_a \models \text{PL} \gamma \). If that would be the case then \( G_a(r, i) \subseteq [\Diamond \gamma] \) and \( B_a(r, i) \subseteq [\gamma] \) which again contradicts (R1).

**Case** \( \text{msg} = ? \psi \): Assume \( \psi \) is not a tautology and \( \Sigma'_a \models_{MS} \psi \). Then, \( \mathfrak{M}, r, i \models B_b B_a \psi \) and hence \( B'_b(r, i) \subseteq [\psi] \), which contradicts (R3).

**Case** \( \text{msg} = ! \psi \): Beliefs are treated as in the previous case. Assume \( \psi \) is not a tautology and \( \Gamma'_a \not\models \psi \). Then also \( \mathfrak{M}, r, i \not\models B_b G_a \Diamond \psi \) which is equivalent to \( \neg (\forall (r', i') \in B_b : (\exists (r'', i'') \in G_b(r', i') : (\mathfrak{M}, r'', i'' \models \psi)) \) and hence \( \exists (r'', i'') \in G_b(r, i) : (\mathfrak{M}, r'', i'' \not\models \Diamond \psi) \). So we have that \( G_b(r, i) \not\subseteq [\Diamond \psi] \) which again contradicts (R3).
Claim 2: \( \forall \phi \in \mathcal{L}_V: f(r), i \models_V \phi \iff \mathcal{M}, t, i \models tr(\phi) \)

This proof is straightforward by induction on \( \phi \) \( \square \)

The theorems show that both the translation from \( \mathcal{L}_V \) to \( tr(\mathcal{L}_V) \) preserves truth given that the modal semantics incorporates principles that match those in the computational transition semantics.

The single agent part of the logic \( \mathcal{L}_M \) extends the logic discussed in [9], apart from a few operators that were not introduced here due to space limits but which may be added without much effort. The result presented here thus extends that of [9] to the multi-agent case including communicative acts\(^3\). In particular, this means that we can relate Cohen and Levesque’s Intention Logic in a similar way to \( \mathcal{L}_V \) as done with \( \mathcal{L}_M \) (modulo communication). We do not claim, however, that the logic presented here corresponds to that of [3].

5.3 Applications of the Modal Logic

Our Selling points: Using Modal techniques (model checking), more expressivity, external point of view “verifying the system”. Analysing the agents. Adding the “truth” layer.

We have introduced the verification logic \( \mathcal{L}_V \) and the modal logic \( \mathcal{L}_M \) which can both be used to reason about agents. Why do we need two logics for the same purpose? We follow the argumentation of [9] and consider the comparison of both logics useful to (1) bridge the gap between agent theory and agent programming; and (2) appreciate the benefit that standard tools for modal logics, e.g. for specification and verification, can be applied to agent programs.

Expressivity, Specification and Verification. The correspondence results show that we can use \( tr(\mathcal{L}_V) \) to describe everything what can be described (and verified) by \( \mathcal{L}_V \). However, it is also clear that the full language \( \mathcal{L}_M \) is much more powerful. For example, we can say that an agent \( a \) believes that another agent \( b \) believes \( \psi \) or \( \neg \psi \), \( B_a(B_b \psi \lor B_b \neg \psi) \), which is not possible with \( \mathcal{L}_V \). Actually, the modal logic is too powerful not every formula makes sense, e.g. \( B_a B_b B_c \psi \). One has the select a reasonable fragment. The extra expressiveness makes a lot of sense for the verification and specification of MAS. It might be used to describe desired properties on the systems and then the designer......

Using Standard Techniques. The use of the modal logic

Characterizing Speech Acts. The logic \( \mathcal{L}_M \) provides us with the means to characterize various communicative actions as instances of particular speech acts. For instance, we can say that an agent \( a \) informs agent \( b \) about \( \phi \) if \( \Box Done_a(\text{send}(a,b,\bullet \phi)) \land K_a \phi \land \neg B_a(B_b \phi \lor B_b \neg \phi) \land \Box B_b B_a \phi \).

\( ^3 \) Apart from the [\( \alpha \)] modality.
6 Conclusion, Related, and Future work

Your Promas and JELIA Paper. Parikh + the things we have...

We introduced a computational semantics for agents that communicate at the knowledge level and a logic to reason about communicating agents that may be used to characterize the message exchanges between agents. Messages are used to reconstruct a model of the sender, which also ensures the autonomy of the receiving agent as that agent’s beliefs and goals are not directly affected while the mental model of the sender still may be used to further the agent’s goals.

[5] seems to shift the burden to implement a speech-act based semantics to the programmer, as they require so-called practical reasoning rules to process received messages. [14] also proposes a shift from sender to receiver in a communication semantics for AgentSpeak(L). The approach is event-driven, however, and does not provide a declarative semantics based on mental models nor a logic to reason about communicative actions.

The expressiveness of the logic to reason about communicating agents is limited compared to other logics that have been proposed [12, 3], and remains an issue for future research, but an advantage of our approach is that it is based on a computational semantics.

Finally, although in the semantics proposed here agents internalize the meaning of messages, there are interesting links with social commitment semantics [2], and it may be useful to integrate the dynamics of commitments at a social level with our agent-based semantics.

References


