Modal Operators with Adaptable Semantics for Multi-agent Systems

S. Borgo

LOA (ISTC-CNR), via Solteri 38, 38100 Trento, Italy
borgo@loa-cnr.it

Abstract. We look at extensions of modal logic for representation and reasoning in the area of multi-agent systems. Building on dynamic logic and Henkin quantifiers, we study an unusual type of operators that present important features for capturing concurrency, independence, collaboration, and coordination between agents. The main goal of this paper is to study the semantics of these operators and to show how it can be adapted to capture different types of agents. The formalism allows a formal comparison of a variety of multi-agent systems.

1 Introduction

For about 20 years we have witnessed an increasing interest in the formal study of *multi-agent systems* (MAS) comprising several entities which present independent and autonomous behaviors. The standard logical machinery has proven to be able to capture (although in a scattered way) many characteristics of (MAS). However, it often requires the coexistence of disparate modalities (dynamic, temporal, epistemic, deontic) in the same language. This strategy is not satisfactory because of the complexity of the logical systems obtained. Furthermore, these logics are hard to compare to the point that the uniformity of the very phenomena at stake is lost in the different formalizations. The goal of this paper is to show that the language we have developed has several natural interpretations which allow us to capture different types of agents while maintaining the very same syntax. We introduce a new type of operators that combine modality with quantification and that we dub *quantificational modal operators*.[1] Our work is not limited to a specific notion of agent (and we are not going to give one), we consider this to be an advantage of our approach. For presentation purposes, we often describe the agents as having some degree of rationality. This is not necessary but it helps in conveying the meaning of the operators. For results on some proof-theoretical aspects see [4].

Structure of the paper. In section 2 we introduce the constant modal operators and in section 3 the Henkin quantifiers. In section 4 we modify these for our purposes. Quantificational (one-column) operators are studied in section 5 and multi-column operators in 6. Section 7 presents examples while section 8 relates this formalism to other logical approaches in the literature.

[1] The name was suggested to us by Daniel Leivant.

2 Basic Modalities for MAS

Our logic is a modification of (Elementary) *Dynamic Logic* (*DL*) [6] as used in MAS. The basic operators, called *constant modal operators*, are modalities indexed by constant identifiers denoting actions (not programs). We use several constant identifiers to isolate even the simplest constant modality so to represent the concurrent activity of the agents. In a system with two agents only, say A_1 and A_2 (taken in this order), our modal operators have the shape of a $2 \times n$ matrix ($n > 0$) where the first row lists the actions performed by A_1 (in the order of their execution) and the second row lists the actions performed by A_2. For instance, $\left[\begin{smallmatrix} c_1 & c_3 \\ c_2 & c_4 \end{smallmatrix} \right]$, with c_i's action identifiers, is a modality corresponding to the transition given by the concurrent execution of action c_1 (by agent A_1) and c_2 (by agent A_2) followed by the concurrent execution of action c_3 (by agent A_1) and c_4 (by agent A_2). That is, each entry of the matrix denotes an action and the combination of these actions characterizes the meaning of the modal operator.

More generally, an operator in the shape of a $k \times n$ matrix is a modality for a system with k agents. It is always assumed that the number of rows in the operators matches the number of agents in the system (as a consequence all the operators in the language have k rows). Also, each agent is associated to the same row in all operators. We now state this more formally.

Let *PropId* be a non-empty countable set, the set of *proposition identifiers*. Let *ActId* be a disjoint non-empty countable set whose elements are called *action identifiers*. These are the individual constants of the language. Formulas are built inductively from proposition identifiers through implication ($\rightarrow$), negation ($\neg$), and the modal operators described below. As usual, we shall make use of the standard conventions for $\wedge, \vee$, and $\leftrightarrow$. Let $A_1, \ldots, A_k$ be the agents in the system.

Let $a_{ij} \in ActId$ (not necessarily distinct), then a *constant modality identifier* for k is a $k \times n$-matrix ($n \geq 1$) $M = \begin{smallmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kn} \end{smallmatrix}$. A *constant modal operator* for k is an expression $[M]$ where M is a constant modality identifier for k.

The set of k-formulas (formulas for short) is the smallest set F satisfying:

I) *PropId* $\subseteq F$ (the elements of *PropId* are called *atomic formulas*)
II) $\neg\varphi$ and $\varphi \rightarrow \psi$ are in F if both φ and ψ are in F
III) $[M]\varphi$ is in F if $[M]$ is a constant modal operator for k and φ is in F

Given a set *Act* of actions, a *k-action* is any $k \times n$ matrix ($n \geq 1$) over *Act*.

A *k-agent Kripke Frame* is a triple $\mathcal{K} = \langle W, Act; R \rangle$ with W a non-empty set *(the set of states)*, *Act* a non-empty set *(the set of actions)*, and R a function mapping k-actions over *Act* to binary relations on W: $R\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} \subseteq W \times W$.

A *k-agent Kripke Structure* is a tuple $\mathcal{M} = \langle W, Act; R, [\![\cdot]\!] \rangle$ where $\langle W, Act; R \rangle$ is a k-agent Kripke frame and $[\![\cdot]\!]$ is a function such that $[\![p]\!] \subseteq W$ for $p \in PropId$ and $[\![a]\!] \in Act$ for $a \in ActId$.

If A_1 performs the action (denoted by) a_1, A_2 the action $a_2, \ldots$, agent A_k the action a_k, we write $\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix}$ for the modal operator describing the evolution of the system due to the concurrent execution of actions $[\![a_1]\!], \ldots, [\![a_k]\!]$ by, respectively, $A_1, \ldots, A_k$. That is, the interpretation of $\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix}$ is the k-action $\begin{matrix} [\![a_1]\!] \\ [\![a_2]\!] \\ \vdots \\ [\![a_k]\!] \end{matrix}$.

The valuation function is extended to multi-column operators as follows: if $[A]$ is a multi-column operator obtained by juxtaposition of operators $[B]$ and $[C]$ (i.e. $[A] = [BC]$), then put $R([\![A]\!]) = R([\![B]\!]) \circ R([\![C]\!])$. In other words, $\left[\!\!\left[\begin{matrix} a_{11} \\ a_{21} \\ \vdots \\ a_{k1} \end{matrix}\right]\!\!\right] =_{def} \begin{matrix} [\![a_1]\!] \\ [\![a_2]\!] \\ \vdots \\ [\![a_k]\!] \end{matrix}$ and $\left[\!\!\left[\begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kn} \end{matrix}\right]\!\!\right] =_{def} \left[\!\!\left[\begin{matrix} a_{11} \\ a_{21} \\ \vdots \\ a_{k1} \end{matrix}\right]\!\!\right] \left[\!\!\left[\begin{matrix} a_{12} \\ a_{22} \\ \vdots \\ a_{k2} \end{matrix}\right]\!\!\right] \ldots \left[\!\!\left[\begin{matrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{kn} \end{matrix}\right]\!\!\right]$ extend function $[\![\cdot]\!]$ to all modality identifiers. Note that we write $[\![M]\!]$ for $[\![\,[M]\,]\!]$.

The truth-value of a formula is defined inductively:

1. Let $p \in \mathit{PropId}$, then $\mathcal{M}, s \models p$ if $s \in [\![p]\!]$
2. $\mathcal{M}, s \models \neg\varphi$ if $\mathcal{M}, s \not\models \varphi$
3. $\mathcal{M}, s \models \varphi \rightarrow \psi$ if $\mathcal{M}, s \not\models \varphi$ or $\mathcal{M}, s \models \psi$
4. $\mathcal{M}, s \models [M]\varphi$ if for all $t \in W$ such that $(s,t) \in R([\![M]\!])$, we have $\mathcal{M}, t \models \varphi$

A *k-agent Kripke model* for a set of formulas Σ in the language is a structure $\mathcal{M}$ for k such that all formulas $\varphi \in \Sigma$ hold in all states of $\mathcal{M}$.

This language is a modification of (Elementary) Dynamic Logic. What really changes is the general perspective. We are no longer using a single constant to describe the evolution of the whole system. Only the combination of all *concurrent actions* provides this information.

Finally, we extend the constant operators by allowing free variables to occur in them. The reason for this choice will become clear later. Let $\Im$ be an environment function from a set of variables *Var* to *Act* and let a modality identifier be any $k \times n$ matrix as before but this time with the less restrictive condition $a_{i,j} \in \mathit{ActId} \cup \mathit{Var}$ (for all relevant indices i, j). The extension of F to include these modalities is trivial. Their interpretation requires the new environment function $\Im$. For the formulas of type $[M]\varphi$ where $[M]$ contains free variables, the interpretation is given by the clause 4. above provided we extend the valuation function by defining, for all $x \in \mathit{Var}$, $[\![x]\!] = \Im(x)$.

3 Henkin Quantifiers

Henkin quantifiers [7, 8] are matrices of quantified variables, e.g. $\begin{pmatrix} \forall x_1 & \exists x_2 & \exists x_3 \\ \exists y_1 & \forall y_2 & \exists y_3 \end{pmatrix}$ and were proposed by Henkin as an extension of first-order logic. Syntactically, these are unary operators as the standard quantifiers. There is no restriction on the number or positions of the quantifiers $\forall$ and $\exists$ in the matrix but no variable may occur more than once in the same operator.

Henkin furnishes a semantic interpretation in terms of game semantics and another (equivalent to the first) using Skolem functions. Let us see the latter through an example. We write $\varphi^{x_1,x_2,\cdots}_{a_1,a_2,\ldots}$ for the first-order formula φ with the free occurrences of x_i replaced by a_i. The formula

$$\begin{pmatrix} \forall x_1 & \exists x_2 & \exists x_3 \\ \exists y_1 & \forall y_2 & \exists y_3 \end{pmatrix} \varphi \tag{1}$$

is *true* in structure $(\mathcal{M}, V)$ if the formula $\forall x_1 y_2\ \varphi^{x_2,\quad x_3,\quad y_1,\ y_3}_{g_1(x_1),\ g_2(x_1),\ g_3,\ g_4(y_2)}$ is true where g_1 and g_2 are obtained by Skolemization from formula $\forall x_1 \exists x_2 x_3 \varphi$, i.e. formula (1) with the second row of the Henkin quantifier erased; and analogously g_3 and g_4 are obtained from formula $\exists y_1 \forall y_2 \exists y_3 \varphi$. Note that one cannot use formula $\exists g_1, g_2, g_3, g_4 \forall x_1, y_2\ \varphi^{x_2,\quad x_3,\quad y_1,\ y_3}_{g_1(x_1),\ g_2(x_1),\ g_3,\ g_4(y_2)}$ since here the choice of g_3 and g_4 is not independent from g_1 and g_2 (a similar problem arises for any permutation of the functions in the prefix $\exists g_1, g_2, g_3, g_4$).

In the game-theoretic semantics a Henkin quantifier (H) is used as the board of a game with k couples of players $(\mathbf{V}_1, \mathbf{F}_1), \ldots, (\mathbf{V}_k, \mathbf{F}_k)$ where $\mathbf{V}_i$ is the ith-*verifier* and $\mathbf{F}_i$ the ith-*falsifier*. The game consists of a set of choices (see below) and the purpose is to assign a truth-value for the formula $(H)\varphi$. For this, pair $(\mathbf{V}_i, \mathbf{F}_i)$ plays a (sub)game on row i of (H) choosing how to instantiate the variables in this row. Since the outcome of the whole game results from the choices made and since the verifiers win if $(H)\varphi$ turns out to be (always) true, the falsifiers win otherwise, we see that $\mathbf{V}_i$ and $\mathbf{F}_i$ play with opposite goals in the subgame i:

- $\mathbf{V}_i$ instantiates variables to obtain an environment in which φ is true;
- $\mathbf{F}_i$ instantiates variables to obtain an environment in which φ is false.

In practice, subgame on row i is a sequence of choices. The players proceed from left to right considering one entry at a time. $\mathbf{V}_i$ chooses whenever there is an existentially quantified variable in the entry, $\mathbf{F}_i$ in the opposite case. It is crucial to note that every move in the subgame of row i is public to $\mathbf{V}_i$ and $\mathbf{F}_i$ only. That is, these choices are *never* known to players in other rows. Finally, formula $(H)\varphi$ is true in $(\mathcal{M}, V)$ if φ is true in any $(\mathcal{M}, V')$ where V' differs from V in as much as it associate all variables occurring in row i of (H) with their values in a play of the corresponding subgame on row i. It is false, otherwise. The reader should convince himself that formula $(H)\varphi$ is true in a model $(\mathcal{M}, V)$ if and only if there exists a *strategy*[2] that guarantee the verifiers to win any play of this game.

Fix a model $(\mathcal{M}, V)$ and let $(\mathbf{V}_1, \mathbf{F}_1), (\mathbf{V}_2, \mathbf{F}_2)$ be players, we now apply game-theoretic semantics to (1). All the players are perfectly aware of the syntactic and semantic components: $\mathcal{M}$, V, the semantic clauses, formula (1). On row 1 the subgame begins with $\mathbf{F}_1$ choosing the value of x_1 (since $\forall x_1$ occurs first), and proceeds with $\mathbf{V}_1$ choosing the first time the value of x_2 and then the

[2] Such a strategy is called a *winning strategy* for $(H)\varphi$ and consists in functions $f_1, \ldots, f_k$, called *choice-functions*, such that if we give f_i the existential variable at stake and the previous choices in row i as arguments, then it returns a value (if any) that V_i can choose to win.

value of x_3. Note that $\mathbf{V}_1$, since moving after $\mathbf{F}_1$, knows the value of x_1 when choosing the value of x_2. Furthermore, $\mathbf{V}_1$ knows the values of both x_1 and x_2 when choosing a value for x_3. On row 2, first $\mathbf{V}_2$ chooses the value of y_1, then $\mathbf{F}_2$ chooses the value of y_2 (knowing what has been chosen for y_1). The subgame finishes with $\mathbf{V}_2$ choosing the value of y_3 (knowing the values of both y_1 and y_2). Clearly, the choices for x_1, x_2, x_3 are made without knowing the values chosen for y_1, y_2, y_3 and vice versa. Once the two subgames are over, the chosen values are used to define environment V' defined by: $V'(x) = V(x)$ for all x not in the Henkin quantifier; $V'(y) = \alpha$ for y in row i and α its value in the i subgame. Finally, the truth-value of (1) is *true* if $\mathbf{V}_1$ and $\mathbf{V}_2$ have a strategy to ensure that for all V' output of a game in (1), φ is true in $\langle \mathcal{M}, V' \rangle$. It is false otherwise.

4 Henkin Quantifiers Revisited

In the game-theoretic semantics of Henkin quantifiers two teams of players instantiate variables to determine an environment for the evaluation of the given formula. Since we are concerned with agents, we modify Henkin's interpretation by assuming that, instead of teams of players, the agents of a multi-agent system are in charge of choosing the variables' values. In other terms, we assume that each agent plays the subgame on its row by instantiating the variables there occurring. The distinction between existential and universal quantifiers is preserved assuming that the agent chooses values with *different aims at different stages.* That is, agent A_i chooses aiming at making the formula true (like $\mathbf{V}_i$ would do) wherever there is an existentially quantified variable, and aiming at making the formula false (like $\mathbf{F}_i$) wherever there is a universally quantified variable.

Unfortunately, this change alone does not do justice of the role of the agents in MAS. Indeed, here the agents can choose (in part) the environment they are in but this is not done through a notion of action. There is an obvious mismatch since the agents' decision abilities are not applied to determine their own actions.

We overcome this problem by moving to the semantics of section 2 where we can pair Henkin quantifiers and modality operators as in the following formula

$$\begin{pmatrix} \forall x_1 & \exists x_2 & \exists x_3 \\ \exists y_1 & \forall y_2 & \exists y_3 \end{pmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} p_0 \tag{2}$$

In this formula we apply Henkin quantifiers to open modal formulas to form sentences. The matching position of the quantified variable in the Henkin quantifier and the free occurrence of the same variable in the modal operator (and so the matching size of the operators) is here crucial. Note that here $\forall$ and $\exists$ range *over actions* since the goal is to instantiate the free variables in the modal operators. We now study these quantifiers in the Kripke semantics approach by taking *Act* as domain of quantification.

From sections 2 and 3, formula (2) is interpreted in two steps. First, we provide a game-theoretic interpretation of $\begin{pmatrix} \forall x_1 & \exists x_2 & \exists x_3 \\ \exists y_1 & \forall y_2 & \exists y_3 \end{pmatrix}$ applied to formula $\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} p_0$ by allowing the agents to independently choose from *Act* the values of the variables in their rows.[3] Let α be the value for z in the game and $\Im'$ be the environment defined by $\Im'(z) = \alpha$ if z occurs in the Henkin quantifier, $\Im'(z) = \Im(z)$ otherwise. The second step consists in the evaluation of formula $\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} p_0$ according to the environment $\Im'$. Informally, this matches the orderly execution by the agents of the actions they have just *planned*.

5 Quantificational Modal Operators

Let *ActId*, *PropId*, and *Var* be as in section 2 with $p_0 \in \mathit{PropId}$. As before, we will informally write "the action a", with $a \in \mathit{ActId}$, to mean the action denoted by a, i.e., the action $[\![a]\!] \in \mathit{Act}$.

Since (2) presents two types of operators with a common structure, we can actually merge them in a unique operator without loss of information by writing formula $\begin{bmatrix} \forall x_1 & \exists x_2 & \exists x_3 \\ \exists y_1 & \forall y_2 & \exists y_3 \end{bmatrix} p_0$. The interpretation of this formula is that of (2): a first step provides instances of the quantified variables in the modality via subgames. A second step provides the evaluation of the modal operator obtained by substituting for each variable in the modality its value in the subgame.

Since we want to model also agents that are *committed to do* some action, we need to allow both action identifiers and quantified variables to occur in an operator. This brings us to the following

Definition 1. *(Quantificational identifiers and operators)*
A quantificational modality identifier *for k is a $k \times n$ matrix ($n > 0$) with each entry containing a constant, a variable, or a quantified variable. A* quantificational (modal) operator *for k is an expression $[M]$ where M is a quantificational modality identifier for k.*
qOP stands for the set of quantificational operators for k (k fixed by the context).

The set F of k-formulas is defined as in section 2 now allowing the bigger class of quantificational operators in clause III). From our discussion, the scope of the modal operator is the formula to which it is applied and the scope of a quantifier in a modal operator is the scope of the modal operator itself.[4] As for Henkin quantifiers, a variable can occur only once in a quantificational operator.

5.1 Henkin's Isolated Agents

This section focuses on the interpretation of quantificational modal operators. Consider formula $\begin{bmatrix} \exists x \\ b \end{bmatrix} p_0$ in a system with agents A_1 and A_2.[5] This formula

[3] The game is a trivial modification of the game described previously. It consists of subgames (as before) where, at each entry, the agent embraces the aim indicated by the occurring quantifier. The agents have no knowledge of the choices made at other rows.

[4] Formally, a quantificational entry, say $\forall x$, stands for the quantified variable $\forall x$ as well as for a bound occurrence of x.

[5] The examples generalize easily to k agents.

holds at a state if agent A_1 can choose an action a such that $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] p_0$ is true at that state. Then, formula $\left[\begin{smallmatrix} \exists x \\ b \end{smallmatrix}\right] p_0$ stands for "agent A_1 can choose an action such that after the concurrent execution of it by A_1 and of b by A_2, p_0 holds". Analogously, formula $\left[\begin{smallmatrix} \forall x \\ b \end{smallmatrix}\right] p_0$ states: "no matter the action chosen by agent A_1, after A_1 has executed it and (concurrently) A_2 has executed b, p_0 holds". The intuition is that all the instances obtained by a choice of A_1 need to be considered to state the truth-value of this formula.

Building on the previous cases, the meaning of the remaining one-column operators is easily determined. The natural reading of $\left[\begin{smallmatrix} \forall x \\ \forall y \end{smallmatrix}\right] p_0$ is: "no matter which action a agent A_1 executes and which action b agent A_2 executes, p_0 holds in the reached states". Formula $\left[\begin{smallmatrix} \exists x \\ \exists y \end{smallmatrix}\right] p_0$ is true if the agents can *independently* choose actions, say a and b, such that $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] p_0$ is true, i.e. "for any choice a made by A_1 and any choice b made by A_2, after A_1 has executed a and A_2 has (concurrently) executed b, p_0 holds". Note that the expressions "for any choice a" and "for all a" characterize different sets of actions. Of the remaining one-column operators, consider $\left[\begin{smallmatrix} \forall x \\ \exists y \end{smallmatrix}\right] p_0$. Since the agents choose independently, i.e. not knowing each other doing, for the formula to be true the second agent has to find an action a for y such that $\left[\begin{smallmatrix} \forall x \\ a \end{smallmatrix}\right] p_0$ is true according to what said above, that is, no matter what the other agent chooses.

To capture formally this interpretation, let us assume that function g furnishes the actions chosen by the agents. The intent is that g codifies the behavior of agent A_i when taking as argument the formula to be evaluated (the modality plus its scope formula) and the variable in row i (which implicitly gives the agent's index); on this input, g returns (one or more) actions in Act which corresponds to agent A_i's choices. Then, the semantics of section 2 is extended with the following clause for quantificational *one-column* operators:[6]

5_1) Given a formula $[X]\varphi$ where $[X]$ is a quantificational operator with variables $x_1, \dots, x_r$ existentially quantified and $y_1, \dots, y_s$ universally quantified; $\mathcal{M}, s.\Im \models [X]\varphi$ if for all given $\alpha_1, \dots, \alpha_r \in Act$ such that $\alpha_i \in g([X], \varphi, x_i)$ and for all $\beta_1, \dots, \beta_s \in Act$, if Γ is the k-action obtained by substituting, in $[X]$, $[\![a_h]\!]$ for $a_h \in ActId \cup Var$, α_i for $\exists x_i$, and β_j for $\forall y_j$ (for all relevant indices h, i, j), then for all $(s, s') \in R(\Gamma)$, $\mathcal{M}, s', \Im' \models \varphi$ with $\Im'$ defined by: $\Im'(x_i) = \alpha_i$, $\Im'(y_j) = \beta_j$, and $\Im'(z) = \Im(z)$ for the remaining cases.

We dub the agents described by this semantic clause *Henkin's isolated agents*; "Henkin's" because of the overall semantics, and "isolated" for the lack of communication among the agents (the above clause prevents the possibility of coordination plans among agents).

[6] Clause 5_1) can be seen as a schema since varying g we capture different MAS.

Formulas $\begin{bmatrix}\exists x\\ \forall y\end{bmatrix} p_0 \rightarrow \begin{bmatrix}\exists x\\ a\end{bmatrix} p_0$; $\begin{bmatrix}a\\ \forall y\end{bmatrix} p_0 \rightarrow \begin{bmatrix}\exists x\\ \forall y\end{bmatrix} p_0$ are valid in this semantics. Interestingly, and perhaps surprisingly, formulas $\begin{bmatrix}\exists x\\ \forall y\end{bmatrix} p_0 \not\rightarrow \begin{bmatrix}\exists x\\ \exists y\end{bmatrix} p_0$ and $\begin{bmatrix}\exists x\\ a\end{bmatrix} p_0 \not\rightarrow \begin{bmatrix}\exists x\\ \exists y\end{bmatrix} p_0$ fail in general. In the first case, for example, let p_0 stand for "have the cake sliced." In the case of $\begin{bmatrix}\forall x\\ \exists y\end{bmatrix} p_0$, agent A_2 would do action "cut the cake" no matter what the other agent does since the latter is not committed to this goal. In the case of $\begin{bmatrix}\exists x\\ \exists y\end{bmatrix} p_0$, agent A_1 (knowing agent A_2 has the same goal) may be polite and let A_2 cut. Similarly, agent A_2 may not do it to let A_1 the honor. Then, nobody cuts the cake (nobody chooses that action) and the formula turns out to be false.

5.2 Risk-averse Coordinated Agents

So far our reading of $\forall$ and $\exists$ was driven by Henkin's work. We now investigate the interpretation one obtains when adopting the classical meaning for $\exists$ and $\forall$. Here we take formula $\begin{bmatrix}\exists x\\ b\end{bmatrix} p_0$ to be true at a state s if and only if there exists an action a such that $\begin{bmatrix}a\\ b\end{bmatrix} p_0$ is true at s. In this reading, the quantificational formula stands for "there exists an action such that after the concurrent execution of it by A_1 and of b by A_2, p_0 holds". Analogously, formula $\begin{bmatrix}\forall x\\ b\end{bmatrix} p_0$ reads: "for any action a, after A_1 has executed it and (concurrently) A_1 has executed b, p_0 holds". It follows easily that formula $\begin{bmatrix}\exists x\\ \exists y\end{bmatrix} p_0$ is true if there exist actions a and b (not necessarily distinct) such that $\begin{bmatrix}a\\ b\end{bmatrix} p_0$ holds, while formula $\begin{bmatrix}\forall x\\ \forall y\end{bmatrix} p_0$ is true if for all actions a and b, $\begin{bmatrix}a\\ b\end{bmatrix} p_0$ is true. An interesting case is given by operators in which both quantifiers occur, for instance $\begin{bmatrix}\forall x\\ \exists y\end{bmatrix} p_0$. To establish the truth value of this formula at a given state we have two choices: either we verify that a value b for y exists such that $\begin{bmatrix}\forall x\\ b\end{bmatrix} p_0$ is a true formula according to our interpretation above. Or we verify that no matter which action a is substituted for x, formula $\begin{bmatrix}a\\ \exists y\end{bmatrix} p_0$ is true in $\mathcal{M}$ at s.

Here we consider the first interpretation only. To capture it formally, we adopt the semantics of section 2 with the following clause for quantificational *one-column* operators:

5_2) Given a formula $[X]\varphi$ where $[X]$ is a quantificational operator with variables $x_1, \ldots, x_r$ existentially quantified and $y_1, \ldots, y_s$ universally quantified; $\mathcal{M}, s, \Im \models [X]\varphi$ if there exist $\alpha_1, \ldots, \alpha_r \in \mathit{Act}$ such that for all $\beta_1, \ldots, \beta_s \in \mathit{Act}$, if Γ is the k-action obtained by substituting, in $[X]$, $[\![a_h]\!]$ for $a_h \in \mathit{ActiId} \cup \mathit{Var}$, α_i for $\exists x_i$, and β_j for $\forall y_j$ (for all relevant indices h, i, j), then for all $(s, s') \in R(\Gamma)$, $\mathcal{M}, s', \Im' \models \varphi$ with $\Im'$ defined by: $\Im'(x_i) = \alpha_i$, $\Im'(y_j) = \beta_j$, and $\Im'(z) = \Im(z)$ in the remaining cases.

We dub the agents satisfying this clause *risk-averse coordinated agents*: "risk-averse" because they never take chances relying on other agents choices, and "coordinated" because they always agree on a combination of actions good for reaching common goals (if any).

In this semantics $\begin{bmatrix} \exists x \\ \forall y \end{bmatrix} p_0 \to \begin{bmatrix} \exists x \\ \exists y \end{bmatrix} p_0$ and $\begin{bmatrix} \exists x \\ a \end{bmatrix} p_0 \to \begin{bmatrix} \exists x \\ \exists y \end{bmatrix} p_0$ are valid.

6 Knowing the Past, Reasoning about the Future

Consider a constant two-column operator[7] $\left[\begin{smallmatrix} c_1 & c_3 \\ c_2 & c_4 \end{smallmatrix}\right]$, call it $[X]$. From clause 4. of section 2, constant operators split into simpler operators without loss of information, that is $\left[\begin{smallmatrix} c_1 & c_3 \\ c_2 & c_4 \end{smallmatrix}\right]\varphi \equiv \left[\begin{smallmatrix} c_1 \\ c_2 \end{smallmatrix}\right]\left[\begin{smallmatrix} c_3 \\ c_4 \end{smallmatrix}\right]\varphi$. This equivalence does not hold for quantificational operators though, i.e. in general

$$[M_1 M_2]\varphi \not\equiv [M_1][M_2]\varphi \quad (M_1, M_2 \in qOP) \tag{3}$$

To establish the truth-value of a formula where the constant operator $[X]$ occurs, it is necessary to consider all action identifiers (and their positions) occurring in $[X]$. For instance, the information that c_1, c_3 occur in the first row in this order (i.e., knowing the actions executed by agent A_1), does not suffice in general to know which states are reachable through $[X]$.

Suppose now that we are dealing with formula $\left[\begin{smallmatrix} c_1 & \exists x \\ c_2 & c_4 \end{smallmatrix}\right] p_0$. It seems natural to read the modality in this formula as follows: "first, agent A_1 executes c_1 and (concurrently) agent A_2 executes c_2, then A_1 chooses and executes an action and (concurrently) A_2 executes c_4." In light of the previous sections, one can interpret the existential quantifier in different ways. Assume for a moment that I am agent A_1 and I am given the model $\mathcal{M}$, the state at which the formula is evaluated, and all the semantic clauses. Then, my choice for x will differ depending on what I *know* about the formula itself and in particular about operator $\left[\begin{smallmatrix} c_1 & \exists x \\ c_2 & c_4 \end{smallmatrix}\right]$. For if I am aware of the presence of c_1, c_2, c_4 and of their positions, I can use the semantic clauses to verify whether there is an action that, when executed by me after the execution of c_1, forces the system to states satisfying p_0. Assuming such an action exists, it might not be possible to identify it if I lack some information about the constant identifiers occurring in the operator.

This argument shows that to establish the truth-value of the formula, it is important to state my knowledge (or lack of) about the operator. Several options are possible. For instance, assuming perfect recall, one can assume that I am aware c_1 is in position (1,1) of the matrix since I have just chosen a value for x. If A_1 and A_2 are *isolated* agents, then we should assume that I (agent A_1) have no idea of what A_2 has done earlier, that is, I do not know what is in position (2,1) of the operator. If A_1 and A_2 are *non-communicating* but can observe each other's doings, one can assume that I know what A_2 has just done, i.e., I know that c_2 is in position (2,1). Finally, for the simple reason that A_1 and A_2 act concurrently, I might know what A_2 is going to execute as second

[7] Our examples use mostly two-column operators. The generalization to n-column operators is generally straightforward.

action only if we are *coordinated* agents or if it is publicly known that agent A_2 has to execute c_4 at this point.[8]

In this paper, we do not enter into the formalization of the semantics for multi-column operators. However, it is possible to extend clauses 5_1) and 5_2) to multi-column operators by allowing $[X]$ to be any operator in qOP. We will make use of this fact in the next section.

7 Modeling with Quantificational Operators

Our first example is in the area of planning. There are two agents, say Anthony (A_1) and Bill (A_2), and they are in charge of a project which should be turned in by a certain time. Let us say that there are 3 time-steps before the deadline (step-1, step-2, and step-3) and that Anthony cannot work at the project at step-1 since he is committed to do something else (perhaps he has to go to the bank or to meet with the company accountant). We use a for the action Anthony does at this step. Later, he is working full time on the project. Regarding Bill, the office manager already asked him to go to his office at the time corresponding to step-2 (but he did not say what the meeting is about). We represent this case in our language with the formula $\left[\begin{smallmatrix} a & \exists x & \exists z \\ \exists y & \forall u & \exists v \end{smallmatrix}\right] \varphi$, where φ stands for "the project is finished". The first row describes Anthony's attitude toward the project during this time, while the second row describes Bill's attitude. Note that the universal quantifier marks the time-step when Bill acts without regards for the project since his action at that time depends on what his office manager asks him to do. Assuming Anthony and Bill are risk-averse cooperative agents, all the actions that instantiate variables x, z, y, v should be planned together as described by clause 5_2) provided it is extended to multi-column quantificational operators as indicated earlier.

We may want to model the case where the agents have a predefine plan for the first two time-steps only. For instance, suppose they discussed a plan the day before when they knew they where going to be in different places during time-steps 1 and 2 without the possibility of sharing information. Also, let us say that later they meet, share they achievements and decide together what to do for the remaining time. This situation is described by formula $\left[\begin{smallmatrix} a & \exists x \\ \exists y & \forall u \end{smallmatrix}\right] \left[\begin{smallmatrix} \exists z \\ \exists v \end{smallmatrix}\right] \varphi$ using clause 5_2) to ensure coordination.

The second example we consider comes from robotics. Here there are two agents whose goal is to pick up an object but none of them can do it alone. If φ stands for "the object is lifted", the situation is described by formula $\left[\begin{smallmatrix} \exists x \\ \exists y \end{smallmatrix}\right] \varphi \wedge \left[\begin{smallmatrix} \forall x \\ \exists y \end{smallmatrix}\right] \neg\varphi \wedge \left[\begin{smallmatrix} \exists x \\ \forall y \end{smallmatrix}\right] \neg\varphi$. The reader can easily verify that this formula is true in 5_2).

[8] In the previous discussion of one-column operators we implicitly assumed that the constant identifiers are known to all the agents, they are *public knowledge*, so to speak. Here we drop this assumption as well. Indeed, one may have a commitment to do a specific action c_i at some point and prevent other agents from knowing it (an issue raised in modeling security). The semantic clauses we have introduced for one-column operators can capture these cases as well.

Of course, these examples can be captured in other formalisms as well, in particular through different languages that include some type of epistemic operators [11]. However, we remark that (*i*) the formulas one obtains in our language are very simple and (*ii*) through our language agents in different systems can be compared immediately by looking at the adopted semantics.

8 Related Work and Conclusions

This paper continues the work presented in [2–4]. In [2] the general approach is given by focusing on the propositional case and its properties. In [3] we studied an interpretation along the lines of 5_1) exploiting fully game-theoretic framework. [4] looks at the formal properties of one of these logics. Differently from these papers, here we have looked at the variety of semantics for our operators and their relationships.

The formalism we adopted has been influenced by the notion of Henkin (branching) quantifiers and their interpretation in game-theory [7]. Nonetheless, branching quantifiers have no modal interpretation and do not allow for semantic alternatives. Furthermore, there is an ontological discrepancy between the notion of agent in MAS (agents are internal components of the system) and the formal notion of player in game-theory (players are external components that act to interpret the formalism).

Somehow related to our work is [1]. The basic features of the logic there presented (without the temporal modalities) are captured through our modal operators using an interpretation similar to 5_2). Our formalism captures *Coalition Logic* [9] as well. Other frameworks, following the *BDI* approach [10] or the *Intention Logic* [5], adopt combinations of different modalities. These are very expressive systems and differ in their motivations from our approach. We refer the reader to [11, 12] for overviews of this area of research.

We have shown how to produce different interpretations for modal operators built out of action identifiers, variables, and quantified variables. Our stand is that when there is a number of practical constraints to capture, semantic pluralism should be sought. In this way, the same descriptive tool can distinguish and characterize different phenomena in a flexible way making possible the uniform description of what might seem a plethora of heterogeneous cases. Then, formal and reliable comparisons of apparently disparate phenomena become possible. Our quantificational modal operators, although limited in several ways, give a first answer to this search for semantic pluralism in the area of multi-agent systems.

Among the features of these quantificational modal operators, the followings are particularly relevant: (*a*) true concurrency is captured already at the syntactic level; (*b*) they can express independence among agents; (*c*) they are naturally associated with different semantics making possible the characterization of different agents; (*d*) they model partial knowledge and communication among agents. There are drawbacks as well. The quantificational operators inherit the restrictions of (Elementary) *DL*, in particular the rigid structure in finite steps. Extensions using constructs on action identifiers have not been studied yet. On the technical side, although adding quantificational modal operators does not make the resulting logic necessarily undecidable, this happens in many cases when equality is present. For instance, one can see that the theory in [3] is undecidable since we can embed first-order logic augmented with a binary predicate (the translation is given by: $A(x, y) \longmapsto [\,^{x}_{y}\,]\, p_0$ for some atomic p_0). For an example in the opposite sense, a slight modification of clause 5_2) gives a complete and decidable logic for the class of multi-relational Kripke frames [4].

Acknowledgments The author has been partially supported by the Provincia Autonoma di Trento. Thanks to Daniel Leivant and Alessandra Carbone for their comments on an earlier draft of this paper.

References

1. R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. In de Roever W.-P., L. H., and P. A., editors, *Compositionality - The Significant Difference*, LNCS 1536, pages 23–60. Springer-Verlag, 1999.
2. S. Borgo. Concurrency with partial information. In M. Mohammadian, editor, *CIMCA '03*, pages 170–181, 2003.
3. S. Borgo. A multi-agent modal language for concurrency with non-communicating agents. In *CEEMAS '03*, LNAI 2691, pages 40–50, 2003.
4. S. Borgo. Quantificational modal logic with sequential kripke semantics. *Journal of Applied Non-Classical Logics*, 15(2):137–188, 2005.
5. P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.
6. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
7. L. Henkin. Some remarks on infinitely long formulas. In *Infinitistic Methods*, pages 167–183. Pergamon Press, Warsaw, 1961.
8. M. Krynicki and M. Mostowski. Henkin quantifiers. In M. Krynicki, M. Mostowski, and S. L.W., editors, *Quantifiers: Logics, Models and Computation*, pages 193–262. Kluwer Academic Publishers, 1995.
9. M. Pauly. A modal logic for coalitional power in games. *J. of Logic and Computation*, 12(1):149–166, 2002.
10. A. Rao and M. Georgeff. Modeling rational agents within a bdi-architecture. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann, 1991.
11. W. van der Hoek and R. Verbrugge. Epistemic logic: a survey. In L. Petrosjan and V. Mazalov, editors, *Game Theory and Applications*, pages 53–94. Nova Science Publishers, 2002.
12. W. van der Hoek and M. J. Wooldridge. Towards a logic of rational agency. *L. J. of the IGPL*, 11(2):135–159, 2003.