Concepts, Agents, and Coalitions in Alternating Time

Wojciech Jamroga[1]

Abstract. We consider a combination of the strategic logic ATL with the description logic $\mathcal{ALCO}$. In order to combine the logics in a flexible way, we assume that every individual can be (potentially) an agent. We also take the novel approach to teams by assuming that a coalition has an identity on its own, and hence its membership can vary. In terms of technical results, we show that the logic does *not* have the finite model property, though both ATL and $\mathcal{ALCO}$ do. We conjecture that the satisfiability problem may be undecidable. On the other hand, model checking of the combined logic is decidable and even tractable. Finally, we define a particular variant of realizability that combines satisfiability of $\mathcal{ALCO}$ with model checking of the ATL dimension, and we show that this new problem is decidable.

1 Introduction

Description logics (DLs) are logical formalisms for representing the knowledge of an application domain in a structured way [4]. More precisely, DLs allow to describe classes, assign individuals to these classes, and define binary relations on individuals. The importance of DLs lies in the fact that comprise the formal basis of the Semantic Web ontology languages [6], and they have well developed practical decision procedures. On the other hand, alternating-time temporal logic (ATL) [3] is probably the most influential logic of strategic ability that has emerged in recent years. In this paper, we propose a product-style combination of ATL with the description logic $\mathcal{ALCO}$. An obvious motivation is to extend ATL with limited first-order component for reasoning about individuals (objects in the world). Alternatively, one can see it as extending the description logic $\mathcal{ALCO}$ with means to explicitly reason about what the actors in the system can achieve over time. We call the resulting language *alternating-time description logic* (ADL).

ADL allows to specify how agents/coalitions can change the properties of the external world, but also how they influence their own characteristics. To some extent this was already present in [10] but that formalism was limited in two respects. First, by using coalition logic (CL) rather than ATL for reasoning about the outcome of agents' actions, specifications were restricted to properties that can be enforced *in the next moment* (or, in a fixed and pre-specified number of steps). This was a serious limitation since most interesting properties refer to patterns that persist over time: either as invariants (specifying e.g. safety conditions of the system) or in terms of reachability (specifying e.g. goals that should be eventually achieved).

Secondly, unlike in [11] where the sets of agents and objects (individuals) were rigidly separated, or in [10] where agents were assumed a special kind of individuals that had to be "called by name", we want to allow here for flexible specifications of coalitions, so that the advantages of description logics are really used. To this end, we assume that coalitions are just concepts, and they are specified in the same way as any other concept. Since concepts are semantically sets of individuals, this suggests that any individual can be an agent, at least potentially. That poses some important semantic issues, as now the number of agents can become infinite. We also take the novel approach to modeling teams by assuming that a coalition has an identity on its own, and hence its membership can vary.

In terms of technical results, we show that the logic is strictly more expressive than ATL, $\mathcal{ALCO}$, and $\mathsf{CL}_{\mathcal{ALCO}}$. On the other hand, it does *not* have the finite model property, though ATL, $\mathcal{ALCO}$, as well as $\mathsf{CL}_{\mathcal{ALCO}}$ do. We conjecture that the satisfiability problem may be undecidable. On the other hand, model checking of the combined logic is decidable and even tractable (in the size of the input). Finally, we define a particular variant of realizability that combines satisfiability of $\mathcal{ALCO}$ with model checking of the ATL dimension, and we show that this new problem is decidable.

2 Preliminaries

We begin by a short presentation of the logics ATL and $\mathcal{ALCO}$.

2.1 Alternating-time Temporal Logic

ATL [3] is a generalization of the branching-time logic CTL, in which path quantifiers are replaced with so called *cooperation modalities* $\langle\langle C\rangle\rangle$. Formula $\langle\langle A\rangle\rangle\gamma$ expresses that coalition A has a collective strategy to enforce property γ. Formally, the language of ATL is given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \langle\langle A\rangle\rangle X\varphi \mid \langle\langle A\rangle\rangle G\varphi \mid \langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\varphi$$

where p is an atomic proposition from a countable set $Prop = \{p_1, p_2, \ldots\}$. Typically, coalitions in ATL formulae are given by listing the names of their members. Cooperation modalities are followed by standard temporal operators: X (next), G (always), $\mathcal{U}$ (until). Additionally, F (sometime in the future) can be defined as: $\langle\langle A\rangle\rangle F\varphi \equiv \langle\langle A\rangle\rangle\top\,\mathcal{U}\,\varphi$.

The semantics of ATL is defined in a variant of transition systems where transitions are labeled with combinations of actions, one per agent. Formally, a *concurrent game structure* (CGS) is a tuple $M = \langle Ag, St, V, Act, d, o\rangle$ which includes a nonempty finite set of all agents $Ag = \{1, \ldots, k\}$, a nonempty set of states (or possible worlds) St, a valuation of atomic propositions $V : Prop \to 2^{St}$, and a nonempty set of (atomic) actions Act. Function $d : Ag \times St \to 2^{Act}$ defines nonempty sets of actions available to agents at each state, and o is a (deterministic) transition function that assigns the outcome state $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to state q and a tuple of actions $\langle\alpha_1, \ldots, \alpha_k\rangle$, $\alpha_i \in d(i, q)$, that can be executed by Ag in q.

A *strategy* of agent a is a conditional plan that specifies what a is going to do in each possible state. Thus, a (memoryless) strategy can be represented by a function $s_a : St \to Act$, such that $s_a(q) \in$

[1] University of Luxembourg, email: wojtek.jamroga@uni.lu

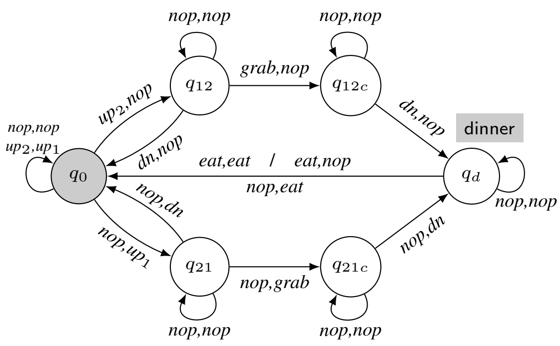

Figure 1. Possums and cheese: a concurrent game structure M_1.

$d(a, q)$. A *collective strategy* for a group of agents $A = \{a_1, ..., a_r\}$ is simply a tuple of strategies $s_A = \langle s_{a_1}, ..., s_{a_r}\rangle$, one per agent from A. We will denote the set of A's collective strategies by Σ_A. Also, by $s_A[a]$, we denote agent a's part of the collective strategy s_A. Function $out(q, s_A)$ returns the set of all paths (i.e., infinite sequences of states) that may occur when coalition A executes strategy s_A from state q onward. The semantics of cooperation modalities is defined below.

$M, q \models \langle\langle A\rangle\rangle X\varphi$ iff there is $s_A \in \Sigma_A$ such that, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda[1] \models \varphi$;
$M, q \models \langle\langle A\rangle\rangle G\varphi$ iff there is $s_A \in \Sigma_A$ such that, for each $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi$ for every $i \geq 0$;
$M, q \models \langle\langle A\rangle\rangle \varphi\mathcal{U}\psi$ iff there is $s_A \in \Sigma_A$ such that, for each $\lambda \in out(q, s_A)$, there is $i \geq 0$ for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for each $0 \leq j < i$.

Example 1 (Dining Possums) *k possums are wandering in an empty house when they spot a beautifully smelling piece of cheese on the table. The table is too high to leap on it, and too slippery to climb, but the possums can reach the cheese if $n \leq k$ of them stand on top of each other. Actions available to possum i are: do nothing (nop), climb on top of another possum j (up_j), climb down (dn), and pick up the cheese ($grab$). An example CGS for $k = n = 2$ is depicted in Figure 1. Note that not every action is enabled in every state. Also, some actions may fail (e.g., when two possums try to climb on each other at the same time). States are named to reflect the configuration, e.g., q_0 is the state where all possums stand on the floor with no cheese, q_{12c} one where possum 1 stands on 2 holding the cheese etc.*

It is easy to see that, in q_0, the possums can cooperate and get to the "dinner" state. However, no possum can get to dinner on its own: $M_1, q_0 \models \langle\langle 1, 2\rangle\rangle F\mathsf{dinner} \wedge \neg\langle\langle 1\rangle\rangle F\mathsf{dinner} \wedge \neg\langle\langle 2\rangle\rangle F\mathsf{dinner}$.

2.2 Description Logic $\mathcal{ALCO}$

Description logics are fragments of monadic first-order logic, widely used as knowledge representation languages. Here, we use $\mathcal{ALCO}$ which extends the standard description logic $\mathcal{ALC}$ with nominals and enumeration of sets [4]. The language of *concepts* and *formulae* of $\mathcal{ALCO}$ is defined as:

$$\begin{array}{rcl} C & ::= & \top \mid C^0 \mid \{\!|\mathsf{i}|\!\} \mid \neg C \mid C \sqcap C \mid \exists R.C \\ \varphi & ::= & C \sqsubseteq C \mid \neg\varphi \mid \varphi \wedge \varphi \end{array}$$

where C^0 represents atomic concepts from a countable set $N_C = \{C_1, C_2, \ldots\}$ of concept names, i is an individual from a countable set N_I, and R a role from a countable set N_R. Intuitively, each concept describes a set of individuals, and concept constructors apply basic set operations on simpler concepts. The following abbreviations can be used: $\bot \equiv \neg\top$, $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$, $\{\!|\mathsf{i}_1 \ldots, \mathsf{i}_n|\!\} \equiv \{\!|\mathsf{i}_1|\!\} \sqcup \cdots \sqcup \{\!|\mathsf{i}_n|\!\}$, $C(\mathsf{i}) \equiv \{\!|\mathsf{i}|\!\} \sqsubseteq C$, and $C \doteq D \equiv (C \sqsubseteq D \wedge D \sqsubseteq C)$.

The semantics is given by a *terminological interpretation* $\mathcal{I} = \langle\Delta, \cdot^{\mathcal{I}}\rangle$, where Δ is a nonempty set called the *domain*, and $\cdot^{\mathcal{I}}$ is a mapping that assigns each atomic concept C_i with a subset $C_i^{\mathcal{I}}$ of Δ, each individual name i with an individual $\mathsf{i}^{\mathcal{I}} \in \Delta$, and each role name R with a binary relation $R^{\mathcal{I}}$ on Δ. The interpretation is extended to other concepts as follows: $\top^{\mathcal{I}} = \Delta$, $\{\!|\mathsf{i}|\!\}^{\mathcal{I}} = \{\mathsf{i}^{\mathcal{I}}\}$, $(\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, and $(\exists R.C)^{\mathcal{I}} = \{\delta \in \Delta \mid \exists\delta' (\langle\delta, \delta'\rangle \in R^{\mathcal{I}} \wedge \delta' \in C^{\mathcal{I}})\}$. Finally, the meaning of formulae is given by $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Example 2 (Possums and Cheese) *Consider the domain* $\Delta = \{pos_1, pos_2, cheese, table\}$, *plus atomic concepts* $\mathsf{Possums}$, Free *and a binary relation* On. *A natural interpretation of the situation in state q_{12} of model M_1 is:* $\mathsf{Possums}^{\mathcal{I}} = \{pos_1, pos_2\}$, $\mathsf{Free}^{\mathcal{I}} = \{pos_1, cheese\}$, $\mathsf{On}^{\mathcal{I}} = \{(pos_1, pos_2), (cheese, table)\}$. *The following formula is true in this interpretation:* $\exists\mathsf{On}.(\mathsf{Possums} \sqcap \mathsf{Free}) = \bot$ *(nothing stands on a free possum).*

3 Concepts and Coalitions in Alternating Time

In this section we introduce our new logic ADL. The logic combines (restricted) first-order features of description logic, and modal approach to reasoning about agents, strategies, and impact of strategic play on evolution of the system. We overcome two limitations that made specification with Coalition Description Logic [10] cumbersome. First, ADL allows for reasoning about long-term temporal patterns (e.g., properties that persist over time). Second, agents and coalitions are treated in ADL like any other individual and concept, which allows for flexible and succinct specification of the interplay between players and their sets.

3.1 Alternating-time Description Logic: Syntax

One way of seeing ADL is that the description logic $\mathcal{ALCO}$ provides concept descriptions and sentences that refer to properties of the current state of the system. The strategic logic ATL adds two kinds of modal operators. Modal *sentence constructors* allow to specify agents' strategic abilities to influence the temporal evolution of the state of the system. Modal *concept constructors* allow to describe the set of individuals that can be influenced in a specified way. We set the sentence constructors in bold to make specifications easier to read.

Formally, the set of concepts is given by the grammar below:

$$C ::= \top \mid C_0 \mid \{\!|\mathsf{i}|\!\} \mid \neg C \mid C \sqcap C \mid \exists R.C \mid \langle\langle C\rangle\rangle XC \mid \langle\langle C\rangle\rangle GC \mid \langle\langle C\rangle\rangle C\,\mathcal{U}\,C,$$

That is, we extend $\mathcal{ALCO}$ concepts with ones referring to the individuals that can be forced by C_1 to join C_2 in the next step ($\langle\langle C_1\rangle\rangle XC_2$), the individuals that can be forced by C_1 to stay in C_2 forever ($\langle\langle C_1\rangle\rangle GC_2$), and so on. We use $\langle\langle C_1\rangle\rangle FC_2$ as the abbreviation for $\langle\langle C_1\rangle\rangle\top\,\mathcal{U}\,C_2$. The set of formulae of ADL is defined as follows (with standard abbreviations):

$$\varphi ::= C \sqsubseteq C \mid \neg\varphi \mid \varphi \wedge \varphi \mid \boldsymbol{\langle\!\langle} C\boldsymbol{\rangle\!\rangle}\mathbf{X}\varphi \mid \boldsymbol{\langle\!\langle} C\boldsymbol{\rangle\!\rangle}\mathbf{G}\varphi \mid \boldsymbol{\langle\!\langle} C\boldsymbol{\rangle\!\rangle}\varphi\,\mathbf{U}\,\varphi.$$

Example 3 (Dining Possums ctd.) *Example* ADL *formulae are:* $\langle\!\langle \mathsf{Hungry} \sqcap \mathsf{Possums} \rangle\!\rangle \mathbf{F}(\mathsf{Hungry} \sqcap \mathsf{Possums} \doteq \bot)$ *(hungry possums can collaborate so that eventually no possum is hungry), and* $(\langle\!\langle \mathsf{Possums} \rangle\!\rangle F \exists \mathsf{In}.\mathsf{Possums}) \doteq \{\!|\mathsf{cheese}|\!\}$ *(cheese is the only object that the possums can eat, i.e., transfer it into a possum).*

Note that $\mathsf{CL}_{\mathcal{ALCO}}$ from [10] can be seen as the "next-time" fragment of ADL, with the additional restriction that coalitions are only specified by enumerating their members.

3.2 How to Interpret Coalitions

In the new syntax, a coalition C is just a concept. This follows the intuition that coalitions are groups of agents, i.e., sets of those individuals who act and influence the evolution of the system. Since we assume that interpretation of concepts can change as the system evolves, the same applies to coalitions. This means in particular that the actual membership in C *may change while the coalition is executing its strategy*. The interpretation has an organizational flavor which is very close to how humans reason about teams. We illustrate this point by a number of examples.

Example 4 (Coalitions evolve over time) *Consider the following statements: "The Rolling Stones (the rock band) have had 9 no. 1 hits to date", "FC Liverpool (the soccer team) can win the next season of Premier League", "the European Union will implement the policy by 2015", "researchers from Malta will keep scoring at least one ECAI paper per decade." In all these cases, we refer to coalitions with potentially varying memberships: the Rolling Stones have been scoring hits with different bassists, drummers and lead guitarists, the EU may enhance or even shrink etc. – yet we do not mean that the policy will be implemented by the current EU states. The statement about Maltese researchers is perhaps the most significant. While the other team descriptions refer to groups that have a clearly established identity (e.g., legal identity), this one is a mere description of a variable set of people. Still, it is interpreted in the same way.*

Another consequence of taking $Ag = \Delta$ is that transitions might be labeled by infinite (and even non-enumerable) tuples of actions. To avoid that, we assume that at each moment only a finite subset of individuals is active. Note that a system can still include infinitely many agents (i.e., acting individuals), but they can only act by taking "turns" of finitely many actions.

3.3 Models

Models, *concurrent game structures with terminological interpretation* (CGSI), are CGS's endowed with limited first-order features. Like in [10], interpretation of concepts and roles can vary from state to state. Also, the domain and interpretation of individual names is assumed to be constant throughout a CGSI.

Definition 1 (CGSI) *A* concurrent game structures with terminological interpretation *is a tuple* $M = \langle \Delta, St, Act, active, d, o, \cdot^{\mathcal{I}(q)} \rangle$*, where:* Δ *is a nonempty domain of interpretation (that defines the set of individuals as well as agents),* St *is a nonempty set of states, and* $active : St \to Fin(\Delta) \setminus \emptyset$ *defines the finite nonempty set of active agents at each state in* St*. Function* $d : St \to (\Delta \rightharpoonup 2^{Act} \setminus \emptyset)$ *defines actions available to agents at particular states; we assume that* $dom(d(q)) = active(q)$*. We will often write* $d_a(q)$ *instead of* $d(q)(a)$*. The transition function* o *defines the next state* q' *given the current state* q *and one action per each active agent in* q*; i.e.,* $o(q, \alpha_{a_1}, \ldots, \alpha_{a_k}) = q'$ *for* $\{a_1, \ldots, a_k\} = active(q)$ *and* $\alpha_{a_i} \in d_{a_i}(q), i = 1, \ldots, k$*. Finally,* $\cdot^{\mathcal{I}(q)}$ *defines the interpretation of atomic concepts and role names for every* $q \in St$*.*

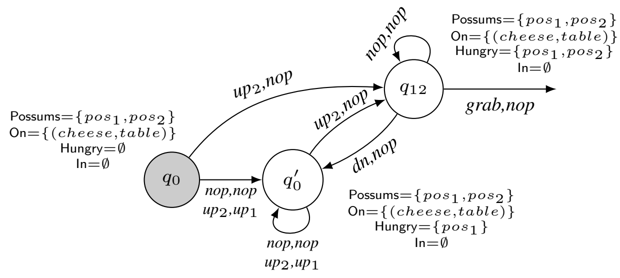

Figure 2. Possums and cheese: a part of the CGSI M_2.

Example 5 (Dining Possums ctd.) *We refine the CGS from Example 1 by the conceptual structure from Example 2, plus an additional concept name* Hungry *to represent the set of hungry individuals and role name* In *for the relation of being inside. We assume that no possum is hungry at the beginning, but it becomes hungry after* t_i *transitions, with* $t_1 = 1$ *for* pos_1 *and* $t_2 = 2$ *for* pos_2*. Moreover, a possum that has just eaten the cheese is not hungry anymore. A part of the resulting CGSI is presented in Figure 2 (we cannot present the whole graph due to lack of space).*

Definition 2 (Frame) *A* concurrent action frame *is a model without interpretation of states, i.e.,* $F = \langle \Delta, St, Act, active, d, o \rangle$*.*

Note that agents/individuals that are never active do not change the action/transition structure. Thus, we say that $M = \langle \Delta', St', Act', active', d', o', \cdot^{\mathcal{I}(q)} \rangle$ extends F *iff* $\Delta \subseteq \Delta'$*, and* $St', Act', active', d', o'$ *are the same as* $St, Act, active, d, o$*.*

3.4 Strategies

Before we give the semantic clauses for ADL, we need to redefine the notion of a collective strategy. In ATL, individual strategies are functions from states to actions, and collective strategies are tuples of individual strategies. However, in our case this would mean that we need to take potentially an infinite number of individual plans, though only finitely many of them would be used at each particular state. To avoid this, we start with the notion of a *joint action* of coalition C: that is, a tuple of actions by the currently active members of C. Formally, the sets of joint actions of C at state q and in the whole system are defined as:

$$d_C(q) = \prod_{\substack{a \in active(q) \\ \cap (C)^{\mathcal{I}(q)}}} d_a(q); \qquad Act_C = \bigcup_{q \in St} d_C(q).$$

Definition 3 (Joint strategy) *A* joint strategy *of coalition* C *is a function* $s_C : St \to Act_C$ *such that* $s_C(q) \in d_C(q)$*. That is,* s_C *prescribes a collective action of* C *in every state* q*. The set of all such strategies is denoted by* $\hat{\Sigma}_C$*.*

Note that we index strategies with *syntactic* rather than semantic entities: C is a concept that describes the coalition, and not its extension! The set of outcome paths of s_C from q on is defined as:

$out(q, s_C) = \{\lambda = q_0 q_1 q_2 \ldots \mid q_0 = q$ and for each $i = 0, 1, 2, \ldots$ there exists a tuple decisions $\langle \alpha_1^i, \ldots, \alpha_k^i \rangle$ for agents in $active(q_i)$ such that $\alpha_a^i \in d_a(q_i)$ for every $a \in active(q_i)$, and $\alpha_a^i = s_C(q_i)[a]$ for every $a \in active(q_i) \cap (C)^{\mathcal{I}(q_i)}$, and $o(q_i, \alpha_1^i, \ldots, \alpha_k^i) = q_{i+1}\}$.

It is easy to see that the new notion of collective strategies coincides with the one from [3] when the membership in C is constant and the members of C are always active (which is the case in ATL):

Proposition 1 *Let $s_C \in \hat{\Sigma}_C$ like in Definition 3, and let $C^{\mathcal{I}(q)} = A \subseteq active(q)$ for all $q \in St$. We construct t_A as the tuple of strategies $t_a : St \to Act$, $a \in A$, such that $t_a(q) = s_C(q)[a]$. Then, $out(q, s_C) = out(q, t_A)$ for every $q \in St$.*

3.5 Interpretation of Concepts and Formulae

We can now finally define the semantics of ADL. We begin with the interpretation of modal concepts:

$(\langle\langle C\rangle\rangle X D)^{\mathcal{I}(q)} = \{x \in \Delta \mid$ there is a joint strategy $s_C \in \hat{\Sigma}_C$ such that for every $\lambda \in out(q, s_C)$ we have $x \in (D)^{\mathcal{I}(\lambda[1])}\}$;
$(\langle\langle C\rangle\rangle G D)^{\mathcal{I}(q)} = \{x \in \Delta \mid$ there is $s_C \in \hat{\Sigma}_C$ such that for every $\lambda \in out(q, s_C)$ and $i = 0, 1, \ldots$ we have $x \in (D)^{\mathcal{I}(\lambda[i])}\}$;
$(\langle\langle C\rangle\rangle D_1 \mathcal{U} D_2)^{\mathcal{I}(q)} = \{x \in \Delta \mid$ there is $s_C \in \hat{\Sigma}_C$ such that for every $\lambda \in out(q, s_C)$ there is i such that $x \in (D_2)^{\mathcal{I}(\lambda[i])}$, and for every $j = 0, \ldots, i-1$ we have $x \in (D_1)^{\mathcal{I}(\lambda[j])}\}$.

It is easy to see that $(\langle\langle C\rangle\rangle F D)^{\mathcal{I}(q)} = \{x \in \Delta \mid$ there is $s_C \in \hat{\Sigma}_C$ such that for every $\lambda \in out(q, s_C)$ there is i such that $x \in (D)^{\mathcal{I}(\lambda[i])}\}$. We also note that the above formulation is more intuitive and easier to read than the one for $\mathsf{CL}_{\mathcal{ALCO}}$ in [11] (compare our clause for $\langle\langle C\rangle\rangle X D$ with the clause for $[C]D$ from that paper).

The semantics of ADL formulae updates that of ATL as follows:

$M, q \models C \sqsubseteq D$ iff $C^{\mathcal{I}(q)} \subseteq D^{\mathcal{I}(q)}$;
$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;
$M, q \models \varphi \wedge \psi$ iff $M, q \models \varphi$ and $M, q \models \psi$;
$M, q \models \langle\!\langle C\rangle\!\rangle \mathbf{X}\varphi$ iff there is a joint strategy $s_C \in \hat{\Sigma}_C$ such that, for each path $\lambda \in out(q, s_C)$, we have $M, \lambda[1] \models \varphi$;
$M, q \models \langle\!\langle C\rangle\!\rangle \mathbf{G}\varphi$ iff there is $s_C \in \hat{\Sigma}_C$ such that, for each $\lambda \in out(q, s_C)$, we have $M, \lambda[i] \models \varphi$ for every $i \geq 0$;
$M, q \models \langle\!\langle C\rangle\!\rangle \varphi \,\mathbf{U}\, \psi$ iff there is $s_C \in \hat{\Sigma}_C$ such that, for each $\lambda \in out(q, s_C)$, there is $i \geq 0$ for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for every $0 \leq j < i$.

Example 6 (Dining Possums ctd.) *An example formula that holds in M_2, q_0 is $\langle\!\langle \mathsf{Hungry} \sqcap \mathsf{Possums}\rangle\!\rangle \mathbf{F}(\{\!\{\mathsf{cheese}\}\!\} \sqsubseteq \exists \mathsf{In}.\mathsf{Possums})$: hungry possums can collaborate so that the cheese is eventually eaten by a possum (note that this requires that the possums first wait so that enough of them become hungry and the coalition* Hungry ⊓ Possums *grows sufficiently). On the other hand, they cannot bring about the cheese having been eaten by a* hungry *possum – as soon as a possum eats, it ceases to be hungry: $M_2, q_0 \not\models \langle\!\langle \mathsf{Hungry} \sqcap \mathsf{Possums}\rangle\!\rangle \mathbf{F}\big(\{\!\{\mathsf{cheese}\}\!\} \sqsubseteq \exists \mathsf{In}.(\mathsf{Hungry} \sqcap \mathsf{Possums})\big)$.*

4 Expressing Interesting Properties with ADL

In this section we discuss what can be expressed in ADL. We start by reworking a motivating example from [10]. We note that, like in ATL, $\langle\langle\bot\rangle\rangle$ and $\langle\langle\top\rangle\rangle$ can be used to express the "for all paths" and "there is a path" quantifiers of branching-time logic.

Example 7 (Authorization) *Let* Perm *stand for the set of permissions to be in a building, and* In *represent the set of agents that are currently inside. Formula $\langle\!\langle\bot\rangle\!\rangle \mathbf{G}(\langle\langle \mathsf{admin}\rangle\rangle F \mathsf{Perm} \doteq \top) \wedge (\langle\langle \mathsf{admin}\rangle\rangle F \neg\mathsf{Perm} \doteq \top)$ specifies that the administrator can grant and deny the permission to any agent. Moreover, $\langle\!\langle\bot\rangle\!\rangle \mathbf{G}(\neg\mathsf{In} \sqcap \langle\langle\top\rangle\rangle X \mathsf{In} \sqsubseteq \mathsf{Perm})$ says that agents who enter the building are only ones that have permission to do so.*

We note three important differences to the specifications from [10]. First, the above formulae specify *invariants* of the system, i.e., properties that will hold at every possible future state. Second, the admin can grant and revoke permissions, but not necessarily in one time step. Third, the above specifications are much more succinct. In particular, they do not require the big conjunction that enumerates all agents by name.

We will now present some general patterns for specification of evolution of concepts, and examine the expressivity of ADL formally.

4.1 General Patterns of Evolution

ADL can capture the following properties of concepts (in particular, coalitions):

- C remains constant in 1 step: $Const_1(C) \equiv (C \sqsubseteq \langle\langle\bot\rangle\rangle X C) \wedge (\langle\langle\top\rangle\rangle X C \sqsubseteq C)$;
- C remains constant throughout every execution of the system: $Const(C) \equiv \langle\!\langle\bot\rangle\!\rangle \mathbf{G} Const_1(C)$;
- $Losless_1(C) \equiv C \sqsubseteq \langle\langle\bot\rangle\rangle X C$ and $Losless(C) \equiv \langle\!\langle\bot\rangle\!\rangle \mathbf{G} Losless_1(C)$ expressing that C does not lose elements (for every possible transition);
- $Grows_1(C) \equiv \neg(\langle\langle\top\rangle\rangle X C \sqsubseteq C)$ and $Grows(C) \equiv \langle\!\langle\bot\rangle\!\rangle \mathbf{G} Grows_1(C)$ stating that there is (always) at least one transition introducing a new element to C.

We will use the last two patterns in Section 5.2 to show that ADL does not have the finite model property.

4.2 Expressive Power of ADL

We use the standard notions of expressive power and distinguishing power [12].

Definition 4 (Distinguishing power) *Logic L_1 is* at least as distinguishing as *L_2 over class of models $\mathcal{M}$ (written $L_2 \leq_d L_1$) iff, for every $M \in \mathcal{M}$ and $\varphi \in L_2$, there is $\varphi' \in L_1$ with the same extension as φ in M. L_1 is* strictly more distinguishing *than L_2 iff $L_2 \leq_d L_1$ but not $L_1 \leq_d L_2$.*

Definition 5 (Expressive power) *Logic L_1 is* at least as expressive *as L_2 over class of models $\mathcal{M}$ (written $L_2 \leq_e L_1$) iff there is a translation $\mathcal{TR}$ from L_2 to L_1 such that for each $\varphi \in L_2$ we have: $M, q \models_{L_2} \varphi$ iff $M, q \models_{L_1} \mathcal{TR}(\varphi)$ for all $M \in \mathcal{M}$ and q in M. L_1 is* strictly more expressive *than L_2 iff $L_2 \leq_e L_1$ but not $L_1 \leq_e L_2$.*

For comparison of distinguishing and expressive power, we observe that $\mathcal{ALCO}$ and $\mathsf{CL}_{\mathcal{ALCO}}$ can be interpreted over CGSI as the "present-time" resp. "next-time" sublanguages of ADL. ATL can be interpreted e.g. by translating atomic propositions p_i to atomic statements $C_i \sqsubseteq \bot$ in ADL (i.e., p_i holds iff the corresponding concept is empty). Moreover, we interpret $\mathcal{ALCO}$ models as single-state ADL models, and ATL models as ADL models where atomic concepts $C_1, C_2, \ldots$ "simulate" propositions $p_1, p_2, \ldots$ so that p_i holds at q iff the interpretation of C_i at q is nonempty. The proofs of the following theorems are rather easy, and we omit them to save space.

Theorem 2 *ADL is strictly more expressive and strictly more distinguishing than ATL as well as $\mathcal{ALCO}$.*

Theorem 3 *ADL is strictly more expressive and strictly more distinguishing than $\mathsf{CL}_{\mathcal{ALCO}}$. In finite models, ADL has the same distinguishing power as $\mathsf{CL}_{\mathcal{ALCO}}$, but strictly more expressive power.*

In short, this is because ADL includes the "transitive closure" operators $F, G, \mathcal{U}$ that can be neither expressed in CL nor simulated in $\mathcal{ALCO}$.

Theorem 4 *The "next-time" fragment of ADL has the same distinguishing power, but strictly more expressive power than $\mathsf{CL}_{\mathcal{ALCO}}$.*

5 Decision Problems and Decidability

5.1 Model Checking

The (global) model checking problem asks, given a finite model M and a formula φ, about the exact set of states $Q \subseteq St_M$ in which φ holds. Below we sketch how ADL model checking can be done by the standard fixpoint model checking algorithm for ATL.

First, we compute the interpretation of concepts in φ by constructing an ATL model M' where points are pairs of states and individuals from M, and concepts play the role of formulae: $St_{M'} = St_M \times \Delta_M$, $Ag_{M'} = \bigcup_{q \in St_M} active_M(q)$, $o_{M'}((q,i), \alpha_1, \ldots, \alpha_k) = \big(o_M(q, \alpha_1, \ldots, \alpha_k), i\big)$,[2] and $V_{M'}(C) = C^{\mathcal{I}_M}$ for all atomic concepts C in φ. We also use $next((q,i), \alpha)$ to denote the set of points in M' that can result from executing a (possibly coalitional) action α from (q,i). For any $\mathcal{C} \subseteq St_M \times \Delta_M$, let $ind(\mathcal{C}, q) = \{i \mid (q,i) \in \mathcal{C}\}$. Now, we define the pre-image function for M' as follows: $pre(\mathcal{C}_1, \mathcal{C}_2) = \{(q,i) \mid \exists \alpha \in d_{M'}\big(ind(\mathcal{C}_1, q), q\big)\ .\ next((q,i), \alpha) \subseteq \mathcal{C}_2\}$. Finally, we use the standard model checking algorithm from [3] with this new function pre to compute the (global) interpretation of all concepts in φ. The algorithm will run in time $O(|St_{M'}| \cdot |\varphi| + |o_{M'}| \cdot |\varphi|) = O(|\Delta_M| \cdot |o_M| \cdot |\varphi|)$.

The final step consists in constructing an ATL model M'' with the same states, actions and transitions as M, and the atomic subformulae of φ treated as atomic propositions: $V(C \sqsubseteq D) = \{q \in St \mid C^{\mathcal{I}_M(q)} \subseteq D^{\mathcal{I}_M(q)}\}$. Now, we simply model check φ in M'' with the algorithm from [3] and return the result. This part of the algorithm will also run in time $O(|\Delta_M| \cdot |o_M| \cdot |\varphi|)$.

Theorem 5 *Model checking ADL is $\mathbf{P}$-complete, and can be done in time linear wrt the number of individuals and transitions in the model, and the length of the formula.*

Proof. $\mathbf{P}$-hardness follows from $\mathbf{P}$-hardness for ATL. Inclusion and the upper bound are guaranteed by the above algorithm. ■

5.2 Satisfiability and Validity

Theorem 6 *ADL does* not *have the finite model property.*

Proof. We recall the patterns from Section 4.1, and observe that $Losless(C) \wedge Grows(C)$ is satisfiable in general but unsatisfiable in finite models (there must be an infinite sequence of states $q_0, q_1, \ldots$ such that each $(C)^{\mathcal{I}(q_i)}$ strictly subsumes $(C)^{\mathcal{I}(q_{i-1})}$). ■

[2] We assume that agents who were not active in M, q are assigned in $M', (q,i)$ only the "no operation" action that does not change the outcome of transitions.

Conjecture 7 *ADL is undecidable.*

Proof idea: we reduce the Halting Problem by simulating a configuration of a Turing machine with a sequence of states that can be "browsed" by agent a_1. Another agent a_2 is responsible for transforming configurations according to transitions of the TM. Then, the TM halts iff a_2 can reach a terminating configuration, with a_1 verifying correctness of transitions on the way.

In the next section we present a more positive result which fits the practice of both DL and MAS communities well.

5.3 Realizability

The communities of description logics and MAS logics differ significantly in what they consider their "standard" decision problems. For description logics, satisfiability is most studied, motivated by the syntactic way in which knowledge bases and ontologies are usually formulated. In contrast, the MAS community mostly studies model checking, because most problem domains can be easily formalized by relational models, often emphasizing the graphical aspect of such modeling. In this section, we propose a variant of realizability[3] that combines model checking in the temporal/strategic dimension with satisfiability of the description logic layer.

Definition 6 (Frame satisfiability) Frame satisfiability *is the decision problem which, given a concurrent action frame F, a state q in it, and a formula φ, answers whether there exists a model M extending F such that $M, q \models \varphi$.*

We will show that frame satisfiability for ADL is decidable. The proof proceeds by a translation of the problem to satisfiability of $\mathsf{CL}_{\mathcal{ALCO}}$, a decidable problem from [10]. The main idea is as follows: (1) We translate the ADL formula φ to a $\mathsf{CL}_{\mathcal{ALCO}}$ formula $tr(\varphi)$ which is equivalent to φ on the given pointed frame F, q_0; (2) We characterize the pointed frame F, q_0 by a $\mathsf{CL}_{\mathcal{ALCO}}$ formula Φ_{F,q_0} that accepts only structures strategically bisimilar [1] to F, q_0; (3) Frame satisfiability for F, q_0, φ is now equivalent to satisfiability of $tr(\varphi) \wedge \Phi_{F,q_0}$. We sketch the formal construction below.

Translation of the formula. Let F, q_0, φ be given (and finite). We will use the following notation: $|F|$ is the number of states in F, $Ag = \bigcup_{q \in St} active(q)$ is the set of all active individuals in F, and $\mathfrak{C} = \{\{\!|\mathsf{i}_1, \ldots, \mathsf{i}_k|\!\} \mid \mathsf{i}_1, \ldots, \mathsf{i}_k \in Ag\}$ the set of all concepts that enumerate possible coalitions in Ag. We will also need an additional relational symbol FC ("fully connected") to facilitate the translation of concepts, and require that all individuals are fully connected to every active agent from Ag by $\mathsf{FC}^{\mathcal{I}}$ (in every state q). This can imposed by the following formula:

$$\Phi_{\mathsf{FC}} \equiv \bigwedge_{i \in Ag} \langle\!\langle \bot \rangle\!\rangle \mathbf{G}(\exists \mathsf{FC}.\{\!|i|\!\}) \doteq \top.$$

Lemma 8 *Let* $\mathrm{ifsub}(A, B, C) \equiv \neg(\exists \mathsf{FC}.(A \sqcap \neg B)) \sqcap C$*. Then, if $\mathsf{FC}^{\mathcal{I}}$ is fully connected in the sense above, and $A^{\mathcal{I}} \subseteq Ag$, we have that $\big(\mathrm{ifsub}(A, B, C)\big)^{\mathcal{I}} = C^{\mathcal{I}}$ if $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ and $\emptyset$ otherwise.*

We translate the subformulae and concepts in φ recursively as below.[4] We use the fact that a successful strategy can be constructed for

[3] The term refers to a class of decision problems where the input is a formula and a *part* of a model. Then, realizability returns "yes" if there exists a model that extends the part and makes the formula true.

[4] For lack of space, we only show translations for the "next" and "always" strategic operators. The case of "until" is analogous, and tr distributes over all the other operators.

$\langle\langle A\rangle\rangle G\varphi, \langle\langle A\rangle\rangle \varphi\, \mathcal{U}\, \psi$ iff there exists a sequence of $|F|$ joint actions for A that preserve the fixpoint translations of $G\varphi, \varphi\,\mathcal{U}\,\psi$ from [3].[5]
$tr(\langle\!\langle C\rangle\!\rangle \mathbf{X}\varphi) = \bigvee_{A\in\mathfrak{C}} \big((A \sqsubseteq tr(C)) \wedge [A]tr(\varphi)\big)$,
$tr(\langle\!\langle C\rangle\!\rangle \mathbf{G}\varphi) = tr_{|F|}(\langle\!\langle C\rangle\!\rangle \mathbf{G}\varphi)$, $\quad tr_0(\langle\!\langle C\rangle\!\rangle \mathbf{G}\varphi) = tr(\varphi)$,
$tr_k(\langle\!\langle C\rangle\!\rangle \mathbf{G}\varphi) = tr(\varphi) \wedge \bigvee_{A\in\mathfrak{C}} \big((A \sqsubseteq tr(C)) \wedge [A]tr_{k-1}(\langle\!\langle C\rangle\!\rangle \mathbf{G}\varphi)\big)$.

By coalitional monotonicity of ATL (if A can enforce γ then also every superset of A can), we have that $\langle\!\langle C\rangle\!\rangle\mathbf{X}$ is about properties that the *current* members C can bring about, while $\langle\!\langle C\rangle\!\rangle\mathbf{G}$ is about the properties that can be preserved by the *subsequent* interpretations of C in $|F|$ steps. The translation of modal concepts is similar, but we need the ifsub macro to obtain the union of outcomes achievable by (subsets of) $C^{\mathcal{I}}$:
$tr(\langle\langle C\rangle\rangle XD) = \sqcup_{A\in\mathfrak{C}}\ \mathrm{ifsub}\big(A, tr(C), [A]tr(D)\big)$,
$tr(\langle\langle C\rangle\rangle GD) = tr_{|F|}(\langle\langle C\rangle\rangle GD)$, $\quad tr_0(\langle\langle C\rangle\rangle GD) = tr(D)$,
$tr_k(\langle\langle C\rangle\rangle GD) = tr(D) \sqcap \big(\sqcup_{A\in\mathfrak{C}}\ \mathrm{ifsub}(A, tr(C), [A]tr_{k-1}(\langle\langle C\rangle\rangle GD))\big)$.

Note that the ifsub construction is not used in $tr(\Phi_{\mathsf{FC}})$. Thus, we obtain the following.

Lemma 9 *For every CGSI M that extends F, let M' be the $\mathsf{CL}_{\mathcal{ALCO}}$ model extending M (by taking the set of agents that collects all active individuals, and setting nop actions for those who do not act). Then, we have that $M, q \models_{\mathsf{ADL}} \varphi$ iff $M', q \models_{\mathsf{CL}_{\mathcal{ALCO}}} tr(\Phi_{\mathsf{FC}}) \wedge tr(\varphi)$.*

Characterizing the frame. Let $AS(\varphi)$ be the set of atomic sentences in φ. We add a concept symbol $State$ that will represent the current state of the frame. Now, we encode the following properties:

- Frame completeness: $\Phi_{comp} \equiv \bigvee_{q\in St_F} State \doteq \{\!|q|\!\}$;
- Uniqueness of state: $\Phi_{uniq} \equiv \bigwedge_{q,q'\in St_F} \neg(\{\!|q|\!\} \doteq \{\!|q'|\!\})$;
- Preservation of atomic properties: $\Phi_{at}(q,\psi) \equiv \big((State \doteq \{\!|q|\!\}) \wedge \psi\big) \to \langle\!\langle\bot\rangle\!\rangle\mathbf{G}\big((State \doteq \{\!|q|\!\}) \to \psi\big)$, $\Phi_{at} \equiv \bigwedge_{q\in St_F} \bigwedge_{\psi\in AS(\varphi)} \Phi_{at}(q,\psi) \wedge \Phi_{at}(q,\neg\psi)$;
- Characterization of one-step transitions. Let $Next^+(q,A) = \{Q \subseteq St_F \mid \exists\alpha \in d_A(q).Q = next(q,\alpha)\}$, i.e., $Next^+(q,A)$ collects all possible outcome sets for A's actions in q. Also, let $Next^-(q,A) = \{Q \subseteq St_F \mid \nexists\alpha \in d_A(q).Q \supseteq next(q,\alpha)\}$ collect outcome sets that cannot be enforced by A.[6] Now: $\Phi_{trans}(q,A) \equiv \bigwedge_{Q\in Next^+(q,A)} \langle\!\langle A\rangle\!\rangle\mathbf{X}(State \sqsubseteq Q) \wedge \bigwedge_{Q\in Next^-(q,A)} \neg\langle\!\langle A\rangle\!\rangle\mathbf{X}(State \sqsubseteq Q)$;
$\Phi_{trans} \equiv \bigwedge_{q\in St_F} \big((State \doteq \{\!|q|\!\}) \to \bigwedge_{A\in\mathfrak{C}} \Phi_{trans}(q,A)\big)$.

Now, formula $\Phi_{F,q_0} \equiv (State \doteq \{\!|q_0|\!\}) \wedge \langle\!\langle\bot\rangle\!\rangle\mathbf{G}(\Phi_{comp} \wedge \Phi_{uniq} \wedge \Phi_{at} \wedge \Phi_{trans})$ can be used to characterize F, q_0 modulo the strategic bisimulation from [1].

Lemma 10 *Let M be a CGSI that extends F with a valuation of $AS(\varphi)$. Then, every pointed model satisfying Φ_{F,q_0} is strategically bisimilar to M, q_0 wrt $AS(\varphi)$. In consequence, all pointed models satisfying Φ_{F,q_0} satisfy the same formulas over $AS(\varphi)$ as M, q_0.*

Wrap-up. By Lemmas 9 and 10, we obtain the following result.

Theorem 11 *ADL formula φ is satisfiable in pointed frame F, q_0 iff the formula $tr(\Phi_{F,q_0}) \wedge tr(\Phi_{\mathsf{FC}}) \wedge tr(\varphi)$ is $\mathsf{CL}_{\mathcal{ALCO}}$-satisfiable.*

Corollary 12 *ADL frame satisfiability is decidable.*

[5] This follows from correctness of the ATL model checking algorithm in [3].
[6] $Next^+(q,A)$ is similar to the nonmonotonic core of A's α-effectivity function in q, and $Next^-(q,A)$ to the complement of the function, cf. [8].

6 Conclusions and Related Work

In this paper, we propose a product-style combination of strategic logic and description logic. We believe that the resulting framework is interesting for at least three reasons. First, it is very expressive and allows for neat and succinct specification of operational concepts that persist in time, or can be enforced through a long-term strategic behavior. Second, we propose a semantics of strategies where the executing team can vary in the runtime – which is a concept very natural for everyday reasoning, but noticeably absent in formal theories of interaction. Thirdly, we propose a variant of realizability that combines traditional approaches to decision problems in the MAS and DL communities, and show that it can be a worthy alternative to both satisfiability and model checking.

Similar ideas have been discussed in a number of papers. A combination of DL and Coalition logic was studied in [11, 10]. Another MAS/DL combination – of the branching-time logic CTL and $\mathcal{ALC}$ – was discussed in [7]. The idea that membership in a coalition can vary throughout execution of a strategy was mentioned in [5], but never formalized nor explored further. Also, referring to coalitions by intensional descriptions was used in [2], but there it was interpreted by a set of (constant) teams, and not a single (but variable) team like in our case. Finally, realizability is a well known problem in temporal logic (cf. e.g. [9]) but the setup is different. There, a part of the temporal structure is usually missing, while we assume that only the valuation of atomic terms and formulae must be synthesized.

Acknowledgements. Wojciech Jamroga acknowledges the support of the FNR (National Research Fund) Luxembourg under project S-GAMES – C08/IS/03.

REFERENCES

[1] T. Ågotnes, V. Goranko, and W. Jamroga, 'Alternating-time temporal logics with irrevocable strategies', in *Proceedings of TARK XI*, ed., D. Samet, pp. 15–24, (2007).
[2] T. Ågotnes, W. van der Hoek, and M. Wooldridge, 'Quantified coalition logic', *Synthese*, **165**(2), 269–294, (2008).
[3] R. Alur, T. A. Henzinger, and O. Kupferman, 'Alternating-time Temporal Logic', *Journal of the ACM*, **49**, 672–713, (2002).
[4] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
[5] N. Bulling and J. Dix, 'Modelling and verifying coalitions using argumentation and ATL', *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, **14**(46), 45–73, (2010).
[6] I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen, 'From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language', *Journal of Web Semantics*, **1**(1), 7–26, (2003).
[7] C. Lutz, F. Wolter, and M. Zakharyaschev, 'Temporal description logics: A survey (with appendix)', in *Proceedings of TIME'08*, (2008).
[8] M. Pauly, *Logic for Social Software*, Ph.D. dissertation, University of Amsterdam, 2001.
[9] A. Pnueli and R. Rosner, 'Distributed reactive systems are hard to synthesize', in *Proceedings of the 31th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 746–757. IEEE Computer Society Press, (1990).
[10] İ. Seylan and W. Jamroga, 'Coalition description logic for individuals', in *Proceedings of the 6th Workshop on Methods for Modalities M4M-6*, pp. 146–162, (2009).
[11] İ. Seylan and W. Jamroga, 'Description logic for coalitions', in *Proceedings of AAMAS'09*, pp. 425–432, (2009).
[12] Y. Wang and F. Dechesne, 'On expressive power and class invariance', *CoRR*, **abs/0905.4332**, (2009).