On Obligations and Abilities

Wojciech Jamroga[1], Wiebe van der Hoek[2], and Michael Wooldridge[2]

[1] Parlevink Group, University of Twente, The Netherlands
jamroga@cs.utwente.nl
[2] Department of Computer Science, University of Liverpool, UK
{wiebe,mjw}@csc.liv.ac.uk

Abstract. In this paper, we combine deontic logic with Alternating-time Temporal Logic (ATL) into a framework that makes it possible to model and reason about obligations *and* abilities of agents. The way both frameworks are combined is technically straightforward: we add deontic accessibility relations to ATL models (concurrent game structures), and deontic operators to the language of ATL (an additional operator $\mathcal{UP}$ is proposed for "unconditionally permitted" properties, similar to the "all I know" operator from epistemic logic). Our presentation is rather informal: we focus on examples of how obligations (interpreted as requirements) can be confronted with ways of satisfying them by actors of the game. Though some formal results are presented, the paper should not be regarded as a definite statement on how logics of obligation and strategic ability must be combined; instead, it is intended for stimulating discussion about such kinds of reasoning, and the models that can underpin it.

Keywords: deontic logic, alternating-time logic, multi-agent systems.

1 Introduction

In recent years, there has been increasing interest from within the computer science, logic, and game theory communities with respect to what might be called *cooperation logics*: logics that make it possible to explicitly represent and reason about the strategic abilities of coalitions of agents (human or computational) in game-like multi-agent scenarios. Perhaps the best-known example of such a logic is the Alternating-time Temporal Logic of Alur, Henzinger, and Kupferman [1]. In this paper, we propose a concept of "deontic ATL". As deontic logic focuses on obligatory behaviors of systems and agents, and Alternating-time Temporal Logic enables reasoning about abilities of agents and teams, we believe it interesting and potentially useful to combine these formal tools in order to confront system requirements (i.e., obligations) with possible ways of satisfying them by actors of the game (i.e., abilities). This paper is not intended as a definite statement on how logics of obligation and strategic ability should be combined. Rather, we intend it to stimulate discussion about such kinds of reasoning, and the models that can underlie it.

We begin by presenting the main concepts from both frameworks. Then, in section 2, their combination is defined and discussed. Three different approaches

A. Lomuscio and D. Nute (Eds.): DEON 2004, LNAI 3065, pp. 165–181, 2004.

to modeling obligations in a temporal context are discussed: global requirements on states of the system (i.e., that deem some states "correct" and some "incorrect"), local requirements on states ("correctness" may depend on the current state), and temporal obligations, which refer to paths rather than states. We investigate (in an informal way) the perspectives offered by each of these approaches, and present several interesting properties of agents and systems that can be expressed within their scope. Some preliminary formal results are given in Section 3.

1.1 Deontic Logic: The Logic of Obligations

Deontic logic is a modal logic of obligations [16], expressed with operator $\mathcal{O}\varphi$ ("it is obligatory that φ"). Models for deontic logic were originally defined as Kripke structures with deontic accessibility relation(s) [21]. A state q' such that $q\mathcal{R}q'$ is called a "perfect alternative" of state q (we can also say that q' is *acceptable* or *correct* from the perspective of q). As with the conventional semantics of modal operators we define,

$$M, q \models \mathcal{O}\varphi \text{ iff for all } q' \text{ such that } q\mathcal{R}q' \text{ we have } M, q' \models \varphi.$$

We believe that this stance still makes sense, especially when we treat deontic statements as referring to preservation (or violation) of some constraints one would like to impose on a system or some of its components (such as integrity constraints in a database). In this sense, deontic modalities may refer to *requirements* (specification requirements, design requirements, security requirements etc.), and we will interpret $\mathcal{O}\varphi$ as "φ is required" throughout the rest of the paper. This approach allows to put all *physically* possible states of the system in the scope of the model, and to distinguish the states that are "correct" with respect to some criteria, thus enabling reasoning about possible faults and fault tolerance of the system [22]. However, we will argue that ATL plus deontic logic allows to express obligations about what coalitions should or should not achieve – without specifying *how* they do achieve it (or refrain from it). We consider this issue in detail in Section 2.5.

Let us illustrate our main ideas with the following example. There are two trains: a and b; each can be inside a tunnel (propositions $\mathsf{a\text{-}in}$ and $\mathsf{b\text{-}in}$, respectively) or outside of it. The specification requires that the trains should not be allowed to be in the tunnel at the same time, because they will crash (so the tunnel can be seen as a kind of critical section): $\mathcal{F}(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$ or, equivalently, $\mathcal{O}\neg(\mathsf{a\text{-}in} \wedge \mathsf{b\text{-}in})$. A model for the whole system is displayed in Figure 1A.

Locality and Individuality of Obligations. Note that the set of perfect alternatives is the same for each state q in the example from Figure 1A. Thus, the semantic representation can in fact be much simpler: it is sufficient to mark the states that *violate* the requirements with a special "violation" atom V [2, 15, 14]. Then the accessibility relation $\mathcal{R}$ can be defined as: $q\mathcal{R}q'$ iff $q' \not\models V$. Using a more elaborate accessibility relation machinery makes it possible, in

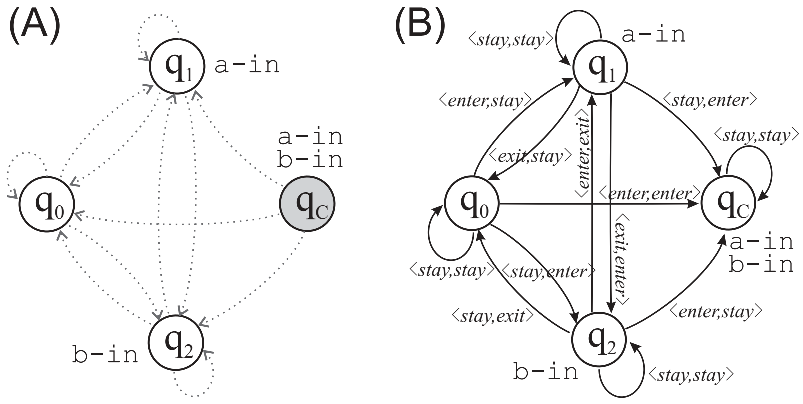

Fig. 1. (A) Critical section example: the trains and the tunnel. Dotted lines display the deontic accessibility relation. (B) The trains revisited: temporal and strategic structure

general, to model requirements that are *local* with respect to the current state. Local obligations can provide a means for specifying requirements that evolve in time. Also, they can be used to specify exception handling in situations when full recovery of the system is impossible (cf. Section 2.3).

Another dimension of classifying obligations is their *individuality*. The accessibility relation can define the requirements for the whole system, or there can be many relations, specifying different requirements for each process or agent [14].

Combining Deontic Perspective with Other Modalities. The combination of deontic logic with temporal and dynamic logics has been investigated at length in the literature [15, 20, 7, 18]. In addition, deontic epistemic logics [5, 14] and BOID ("beliefs-obligations-intentions-desires") logics [6] have also been studied. Finally, in [19], deontic and strategic perspectives were combined through applying social laws to ATL.

1.2 Strategic Ability: Alternating-Time Temporal Logic

Alternating-time Temporal Logic (ATL) [1] extends the computation tree logic (CTL) with a class of *cooperation modalities* of the form $\langle\langle A\rangle\rangle$, where A is a set of agents. The intuitive interpretation of $\langle\langle A\rangle\rangle\Phi$ is: "The group of agents A have a collective strategy to enforce Φ no matter what the other agents in the system do". The recursive definition of ATL formulas is:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\langle A\rangle\rangle X\varphi \mid \langle\langle A\rangle\rangle G\varphi \mid \langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$$

The "sometime" operator F can be defined as: $\langle\langle A\rangle\rangle F\varphi \equiv \langle\langle A\rangle\rangle \top \mathcal{U} \varphi$.

Models and Semantics of ATL. *Concurrent game structures* are transition systems that are based on the collective actions of all agents involved. Formally, a *concurrent game structure* is a tuple $M = \langle \Sigma, Q, \Pi, \pi, Act, d, \delta \rangle$, where:

$\Sigma = \{a_1, ..., a_k\}$ is a (finite) set of all *agents*, Q is a non-empty set of *states*, Π is a set of (atomic) *propositions*, and $\pi : Q \to 2^{\Pi}$ is a *valuation* of propositions; Act is a set of *actions* (or choices), and $d : Q \times \Sigma \to 2^{Act}$ is a function that returns the decisions available to player a at state q. Finally, a complete tuple of decisions $\langle \alpha_1, ..., \alpha_k \rangle \subseteq d_q(a_1) \times \ldots \times d_q(a_k)$ from all the agents in state q implies a deterministic transition according to the transition function $\delta(q, \alpha_1, ..., \alpha_k)$ [1].

A *strategy* for agent a is a mapping $f_a : Q^+ \to Act$, which assigns a choice $f_a(q_0, ..., q_n) \in d_a(q_n)$ to every non-empty finite sequence of states $q_0, ..., q_n$. Thus, the function specifies a's decisions for every possible (finite) history of system transitions. A *collective strategy* for a set of agents $A \subseteq \Sigma$ is just a tuple of strategies (one for each agent in A): $F_A = \langle f_a \rangle_{a \in A}$. Now, $out(q, F_A)$ denotes the *set of outcomes* of F_A from q, i.e., the set of all (infinite) computations starting from q, in which group A has been using F_A. Let $\Lambda[i]$ denote the ith position in computation Λ. The semantics of ATL formulas follows through the clauses:

$M, q \models \langle\langle A \rangle\rangle X\varphi$ iff there exists a collective strategy F_A such that for all $\Lambda \in out(q, F_A)$ we have $M, \Lambda[1] \models \varphi$;

$M, q \models \langle\langle A \rangle\rangle G\varphi$ iff there exists a collective strategy F_A such that for all $\Lambda \in out(q, F_A)$ 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 exists a collective strategy F_A such that for all $\Lambda \in out(q, F_A)$ there is $i \geq 0$ such that $M, \Lambda[i] \models \psi$ and for all j such that $0 \leq j < i$ we have $M, \Lambda[j] \models \varphi$.

Let us consider the tunnel example from a temporal (and strategic) perspective; a concurrent game structure for the trains and the tunnel is shown in Figure 1B. Using ATL, we have that $\langle\langle \Sigma \rangle\rangle F(\textsf{a-in} \wedge \textsf{b-in})$, so the system is physically able to display undesirable behavior. On the other hand, $\langle\langle a \rangle\rangle G\neg(\textsf{a-in} \wedge \textsf{b-in})$, i.e., train a can protect the system from violating the requirements. In this paper, we propose to extend ATL with deontic operator $\mathcal{O}$ in order to investigate the interplay between agents' abilities and requirements they should meet.

The Full Logic of ATL*. ATL* generalizes ATL in the same way as CTL* generalizes CTL: we release the syntactic requirement that every occurrence of a temporal operator must be preceded by exactly one occurrence of a cooperation modality. ATL* consists of *state formulas* φ and *path formulas* ψ, defined recursively below:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\langle A \rangle\rangle \psi$$
$$\psi := \varphi \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid X\psi \mid \psi_1 \mathcal{U} \psi_2$$

Temporal operators F and G can be defined as: $F\psi \equiv \top \mathcal{U} \psi$ and $G\psi \equiv \neg F \neg \psi$. ATL* has strictly more expressive power than ATL, but it is also more computationally costly. Therefore ATL is more important for practical purposes. For semantics and extensive discussion, we refer the reader to [1].

[1] The definition we use here differs slightly from the original one [1], because we use symbolic labels for agents and their choices (and we do not assume finiteness of Q and Act). For an extensive discussion of various ATL semantics, refer to [9].

1.3 STIT Logic: The Logic of Causal Agency

It is also worth mentioning at this point a related body of work, initiated largely through the work of Belnap and Perloff, on "stit" logic – the logic of *seeing to it that* [4, 3]. Such logics contain an *agentive* modality, which attempts to capture the idea of an agent *causing* some state of affairs. This modality, typically written $[i \text{ stit } \phi]$, is read as "agent i sees to it that ϕ". The semantics of stit modalities are typically given as $[i \text{ stit } \phi]$ iff i makes a choice c, and ϕ is a necessary consequence of choice c (i.e., ϕ holds in all futures that could arise through i making choice c). A distinction is sometimes made between the "generic" stit modality and the *deliberate* stit modality ("dstit"); the idea is that i deliberately sees to it that ϕ if $[i \text{ stit } \phi]$ and there is at least one future in which ϕ does not hold (the intuition being that i is then making a *deliberate choice* for ϕ, as ϕ would not necessarily hold if i did not make choice c). Such logics are a natural counterpart to deontic logics, as it clearly makes sense to reason about the obligations that an agent has in the context of the choices it makes and the consequences of these choices. Similarly, if we interpret choices as programs (cf. the strategies of ATL), then stit logics are also related to dynamic logic [12]; the main differences are that programs, which are first class entities in the object language of dynamic logic, are not present in the object language of stit logics (and of course, strategies are not present in the object language of ATL). Moreover, stit logics assert that an agent *makes* a particular choice, whereas we have no direct way of expressing this in ATL (or, for that matter, in dynamic logic). So, while stit logics embody somewhat similar concerns to ATL (and dynamic logic), the basic constructs are fundamentally different, providing (yet another) way of interpreting the dynamic choice structures that are common to these languages.

2 Deontic ATL

In this section, we extend ATL with deontic operators. We follow the definition with an informal discussion on how the resulting logic (and its models) can help to investigate the interplay between agents' abilities and requirements that the system (or individual agents) should meet.

2.1 Syntax and Semantics

The combination of deontic logic and ATL proposed here is technically straightforward: the new language consists of both deontic and strategic formulas, and models include the temporal transition function and deontic accessibility relation as two independent layers. Thus, the recursive definition of DATL formulas is:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathcal{O}_A\varphi \mid \mathcal{UP}_A\varphi \mid \langle\langle A\rangle\rangle X\varphi \mid \langle\langle A\rangle\rangle G\varphi \mid \langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$$

where $A \subseteq \Sigma$ is a set of agents. Models for Deontic ATL can be called *deontic game structures*, and defined as tuples $M = \langle \Sigma, Q, \Pi, \pi, Act, d, \delta, \mathbb{R} \rangle$, where:

- Σ is a (finite) set of all *agents*, and Q is a non-empty set of *states*,
- Π is a set of (atomic) *propositions*, and $\pi : Q \to 2^{\Pi}$ is their *valuation*;
- *Act* is a set of actions, and $d : Q \times \Sigma \to 2^{Act}$ is a function that returns the decisions available to player a at state q;
- a complete tuple of decisions $\langle \alpha_1, ..., \alpha_k \rangle \subseteq d_q(a_1) \times ... \times d_q(a_k)$ from all the agents in state q implies a deterministic transition according to the transition function $\delta(q, \alpha_1, ..., \alpha_k)$;
- finally, $\mathbb{R} : 2^{\Sigma} \to 2^{Q \times Q}$ is a mapping that returns a deontic accessibility relation $\mathcal{R}_A$ for every group of agents A.

The semantic rules for $p, \neg\varphi, \varphi \vee \psi, \langle\langle A \rangle\rangle X\varphi, \langle\langle A \rangle\rangle G\varphi, \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$ are inherited from the semantics of ATL (cf. Section 1.2), and the truth of $\mathcal{O}_A\varphi$ is defined below. We also propose a new deontic operator: $\mathcal{UP}\varphi$, meaning that "φ is unconditionally permitted", i.e., whenever φ holds, we are on the correct side of the picture (which closely resembles the "only knowing"/"all I know" operator from epistemic logic [13]).

$$M, q \models \mathcal{O}_A\varphi \text{ iff for every } q' \text{ such that } q\mathcal{R}_A q' \text{ we have } M, q' \models \varphi;$$
$$M, q \models \mathcal{UP}_A\varphi \text{ iff for every } q' \text{ such that } M, q' \models \varphi \text{ we have } q\mathcal{R}_A q'.$$

This new operator – among other things – will help to characterize the *exact* set of "correct" states, especially in the case of local requirements, where the property of a state being "correct" depends on the current state of the system.

In principle, it should be possible that the requirements on a group of agents (or processes) are independent from the requirements for the individual members of the group (or its subgroups). Thus, we will not assume any specific relationship between relations $\mathcal{R}_A$ and $\mathcal{R}_{A'}$, even if $A' \subseteq A$. We propose only that a system can be identified with the complete group of its processes, and therefore the requirements on a system as a whole can be defined as: $\mathcal{O}\varphi \equiv \mathcal{O}_{\Sigma}\varphi$. In a similar way: $\mathcal{UP}\varphi \equiv \mathcal{UP}_{\Sigma}\varphi$.

2.2 Dealing with Global Requirements

Let us first consider the simplest case, i.e., when the distinction between "good" and "bad" states is global and does not depend on the current state. Deontic game structures can in this case be reduced to concurrent game structures with "violation" atom V that holds in the states that violate requirements. Then:

$$M, q \models \mathcal{O}\varphi \text{ iff for all } q' \text{ such that } q' \not\models V \text{ we have } M, q' \models \varphi.$$

As we have both requirements and abilities in one framework, we can look at the former and then ask about the latter. Consider the trains and tunnel example from Figure 1B, augmented with the requirements from Figure 1A (let us also assume that these requirements apply to all the agents and their groups, i.e., $\mathcal{R}_A = \mathcal{R}_{A'}$ for all $A, A' \subseteq \Sigma$; we will continue to assume so throughout the rest of the paper, unless explicitly stated). As already proposed, the trains are required not to be in the tunnel at the same moment, because it would result in a crash:

$\mathcal{O}(\neg(\textsf{a-in} \wedge \textsf{b-in}))$. Thus, it is natural to ask whether some agent or team can prevent the trains from crashing: $\langle\langle A\rangle\rangle G\neg(\textsf{a-in} \wedge \textsf{b-in})$? Indeed, it turns out that both trains have this ability: $\langle\langle a\rangle\rangle G\neg(\textsf{a-in}\wedge\textsf{b-in})\wedge\langle\langle b\rangle\rangle G\neg(\textsf{a-in}\wedge\textsf{b-in})$. On the other hand, if the goal of a train implies that it passes the tunnel, the train is unable to "safeguard" the system requirements any more: $\neg\langle\langle a\rangle\rangle\neg(\textsf{a-in}\wedge\textsf{b-in})\,\mathcal{U}(\textsf{a-in}\wedge\neg\textsf{b-in})$.

In many cases, it may be interesting to consider questions like: does an agent have a strategy to always/eventually fulfill the requirements? Or, more generally: does the agent have a strategy to achieve his goal in the way that does not violate the requirements (or so that he can recover from the violation of requirements eventually)? We try to list several relevant properties of systems and agents below:

1. the system is *stable* (with respect to model M and state q) if $M, q \models \langle\langle\varnothing\rangle\rangle G\neg V$, i.e., no agent (process) can make it crash;
2. the system is *semi-stable* (with respect to model M and state q) if it will inevitably recover from any future situation: $M, q \models \langle\langle\varnothing\rangle\rangle G\langle\langle\varnothing\rangle\rangle F\neg V$;
3. agents A form a (collective) *guardian* in model M at state q if they can protect the system from any violation of the requirements: $M, q \models \langle\langle A\rangle\rangle G\neg V$;
4. A can *repair the system* in model M at state q if $M, q \models \langle\langle A\rangle\rangle F\neg V$;
5. A is a (collective) *repairman* in model M at state q if A can always repair the system: $M, q \models \langle\langle\varnothing\rangle\rangle G\langle\langle A\rangle\rangle F\neg V$;
6. finally, another (perhaps the most interesting) property is agents' ability to eventually achieve their goal (φ) without violating the requirements. We say that agents A can *properly enforce* φ in M, q if $M, q \models \langle\langle A\rangle\rangle(\neg V)\,\mathcal{U}(\neg V \wedge \varphi)$.

We will illustrate the properties with the following example. The world is in danger, and only the Prime Minister (p) can save it through giving a speech at the United Nations session and revealing the dangerous plot that threatens the world's future. However, there is a killer (k) somewhere around who tries to murder him before he presents his speech. The Prime Minister can be hidden in a bunker (proposition $\textsf{pbunk}$), moving through the city ($\textsf{pcity}$), presenting the speech ($\textsf{pspeaks} \equiv \textsf{saved}$), or... well... dead after being murdered ($\textsf{pdead}$). Fortunately, the Minister is assisted by James Bond (b) who can search the killer out and destroy him (we are very sorry – we would prefer Bond to arrest the killer rather than do away with him, but Bond hardly works this way...). The deontic game structure for this problem is shown in Figure 2. The Prime Minister's actions have self-explanatory labels (*enter, exit, speak* and *nop* for "no operation" or "do nothing"). James Bond can defend the Minister (action *defend*), look for the killer (*search*) or stay idle (*nop*); the killer can either shoot at the Minister (*shoot*) or wait (*nop*). The Minister is completely safe in the bunker (he remains alive regardless of other agents' choices). He is more vulnerable in the city (can be killed unless Bond is defending him at the very moment), and highly vulnerable while speaking at the UN (the killer can shoot him to death even if Bond is defending him). James Bond can search out and destroy the killer in a while (at any moment). It is required that the world is saveable ($\mathcal{O}\langle\langle\Sigma\rangle\rangle F\textsf{saved}$) and this is the only requirement ($\mathcal{UP}\langle\langle\Sigma\rangle\rangle F\textsf{saved}$). Note also that the world can be saved if, and only if, the Prime Minister is alive

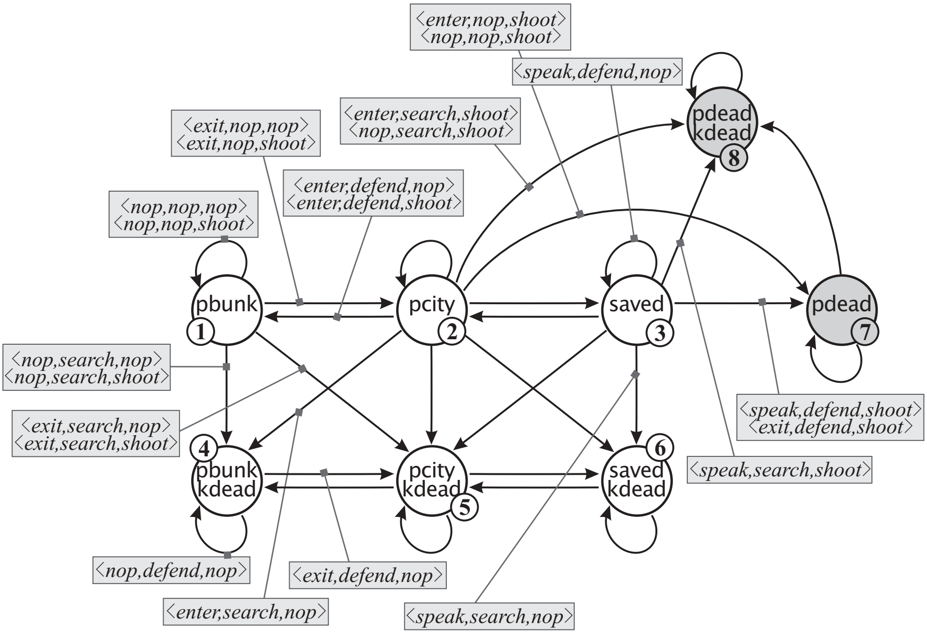

Fig. 2. James Bond saves the world. The arrows show possible transitions of the system; some of the labels are omitted to improve readability. The states that violate the requirements are marked grey

($\langle\!\langle \Sigma \rangle\!\rangle F \mathsf{saved} \equiv \neg \mathsf{pdead}$), and the two states that violate this requirement are marked accordingly ($V \equiv \mathsf{pdead}$).

The system is neither stable nor semi-stable (the Minister can go to the UN building and get killed, after which the system has no way of recovering). Likewise, no agent can repair the system in states q_7, q_8, and hence there is no repairman. The Prime Minister is a guardian as long as he is in the bunker, because he can stay in the bunker forever: $\mathsf{pbunk} \to \langle\!\langle p \rangle\!\rangle G \neg \mathsf{pdead}$. However, if he does so, he cannot save the world: $\neg \langle\!\langle p \rangle\!\rangle (\neg \mathsf{pdead}) \mathcal{U} (\neg \mathsf{pdead} \wedge \mathsf{saved})$. On the other hand, he can cooperate with Bond to properly save the world as long as he is initially out of the UN building: $(\mathsf{pbunk} \vee \mathsf{pcity}) \to \langle\!\langle p, b \rangle\!\rangle (\neg \mathsf{pdead}) \mathcal{U} (\neg \mathsf{pdead} \wedge \mathsf{saved})$ – he can get to the bunker, defended by Bond, and then wait there until Bond finds the killer; then he can go out to present his speech. Incidentally, there is one more guardian in the system – namely, the killer: $(\neg \mathsf{pdead}) \to \langle\!\langle k \rangle\!\rangle G \neg \mathsf{pdead}$, and also $(\neg \mathsf{pdead}) \to \langle\!\langle p, k \rangle\!\rangle (\neg \mathsf{pdead}) \mathcal{U} (\neg \mathsf{pdead} \wedge \mathsf{saved})$, so the Minister can alternatively pay the killer instead of employing Bond.

2.3 Local Requirements with Deontic ATL

A more sophisticated deontic-accessibility relation may be convenient for modeling dynamics of obligations, for instance when the actors of the game can negotiate the requirements (e.g., deadlines for a conference submission). Alter-

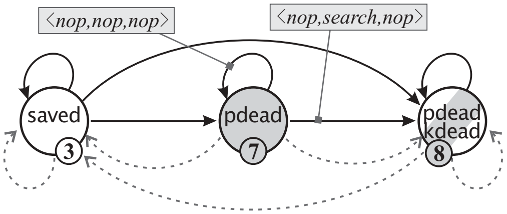

Fig. 3. "James Bond saves the world" revisited: local requirements. Dotted lines define the deontic accessibility relation. Solid lines show possible transitions of the system

natively, "localized" requirements can give a way of specifying *exception handling* in situations when a full recovery is impossible.

Consider the modified "James Bond" example from Figure 3. The Prime Minister is alive initially, and it is required that he should be protected from being shot: $q_3 \models \neg\mathsf{pdead}$ and $q_3 \models \mathcal{O}\neg\mathsf{pdead}$. On the other hand, nobody except the killer can prevent the murder: $q_3 \models \langle\!\langle k\rangle\!\rangle G\neg\mathsf{pdead} \wedge \neg\langle\!\langle p,b\rangle\!\rangle G\neg\mathsf{pdead}$; moreover, when the president is dead, there is no way for him to become alive again ($\mathsf{pdead} \to \langle\!\langle\varnothing\rangle\!\rangle G\mathsf{pdead}$). Now, when the Minister is shot, a new requirement is implemented, namely it is required that either the Minister is resurrected or the killer is eliminated: $q_7 \models \mathcal{O}(\neg\mathsf{pdead} \vee \mathsf{kdead})$. Fortunately, Bond can bring about the latter: $q_7 \models \langle\!\langle b\rangle\!\rangle F\mathsf{kdead}$. Note that q_8 is unacceptable when the Minister is alive (q_3), but it becomes the only option when he has already been shot $(q_7)^2$.

Similar properties of agents and systems to the ones from the previous section can be specified:

1. the system is *stable* in M, q if, given $M, q \models \mathcal{O}p \wedge \mathcal{UP}p$, we have $M, q \models \langle\!\langle\varnothing\rangle\!\rangle Gp$;

2. the system is *semi-stable* in M, q if, given that $M, q \models \mathcal{O}p \wedge \mathcal{UP}p$, we have $M, q \models \langle\!\langle\varnothing\rangle\!\rangle G(p \to \langle\!\langle\varnothing\rangle\!\rangle Fp)$;

3. A form a *guardian* in M, q if, given $M, q \models \mathcal{O}p \wedge \mathcal{UP}p$, we have $M, q \models \langle\!\langle A\rangle\!\rangle Gp$;

4. A can *repair* the system in M, q if, given that $M, q \models \mathcal{O}p \wedge \mathcal{UP}p$, we have $M, q \models \langle\!\langle A\rangle\!\rangle Fp$;

5. group A is a *repairman* in M, q if, given that $M, q \models \mathcal{O}p \wedge \mathcal{UP}p$, we have $M, q \models \langle\!\langle\varnothing\rangle\!\rangle G\langle\!\langle A\rangle\!\rangle Fp$;

6a. A can *properly enforce* φ in M, q if, given that $M, q \models \mathcal{O}_A p \wedge \mathcal{UP}_A p$, we have $M, q \models \langle\!\langle A\rangle\!\rangle p\,\mathcal{U}(p \wedge \varphi)$. Note that this requirement is individualized now;

6b. A can *properly (incrementally) enforce* φ in M, q if, given that $M, q \models \mathcal{O}_A p \wedge \mathcal{UP}_A p$, we have $M, q \models p \wedge \varphi$, or $M, q \models p$ and A have a collective strategy F_A such that for every $\lambda \in out(q, F_A)$ they can properly (incrementally) enforce φ in $M, \lambda[1]$.

[2] In a way, we are making the deontic accessibility relation "serial" in a very special sense, i.e., every state has at least one *reachable* perfect alternative now.

The definitions show that many interesting properties, combining deontic and strategic aspects of systems, can be defined using semantic notions. However, at present, we do not see how they can be specified entirely in the object language.

2.4 Temporal Requirements

Many requirements have a temporal flavor, and the full language of ATL* allows to express properties of temporal paths as well. Hence, it makes sense to look at DATL^*, where one specifies deontic temporal properties in terms of correct computations (rather than single states). In its simplest version, we obtain DTATL by only allowing requirements over temporal (path) subformulas that can occur within formulas of ATL:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\langle A\rangle\rangle\psi \mid \mathcal{O}_A\psi \mid \mathcal{UP}_A\psi$$

with the path subformulas ψ defined recursively as

$$\psi := X\varphi \mid G\varphi \mid \varphi_1\mathcal{U}\varphi_2 \qquad \text{(where } \varphi \in \mathsf{DTATL}\text{)}.$$

Properties that can be expressed in this framework are, for instance, that $\mathcal{O}F\langle\langle\Gamma\rangle\rangle G\varphi$ (it is required that sometime in the future, coalition Γ gets the opportunity to guarantee φ forever) and $\mathcal{O}F(\langle\langle\Gamma\rangle\rangle F\varphi \wedge \langle\langle\Gamma\rangle\rangle F\neg\varphi)$ (it is a requirement that eventually coalition Γ can determine φ). The latter can be strengthened to

$$\mathcal{O}G(\langle\langle\Gamma\rangle\rangle F\varphi \wedge \langle\langle\Gamma\rangle\rangle F\neg\varphi)$$

saying that it is an obligation of the system that there must always be opportunities for Γ to toggle φ as it wants. Note that the definition of DTATL straightforwardly allows to express stability properties like

$$\mathcal{O}T\psi \rightarrow \langle\langle\Gamma\rangle\rangle T\psi$$

saying that Γ can bring about the temporal requirement $T\psi$.

Semantically, rather than being a relation between states, relation $\mathcal{R}_A$ is now one between states and computations (sequences of states). Thus, for any computation λ, $q\mathcal{R}_A\lambda$ means that λ is an ideal computation, given q. The semantics of temporal obligations and unconditional permissions can be defined as:

$M, q \models \mathcal{O}_A X\varphi$ iff for every λ such that $q\mathcal{R}_A\lambda$, we have $M, \lambda[1] \models \varphi$;
$M, q \models \mathcal{O}_A G\varphi$ iff for each λ such that $q\mathcal{R}_A\lambda$, we have $M, \lambda[i] \models \varphi$ for all $i \geq 0$;
$M, q \models \mathcal{O}_A \varphi\mathcal{U}\psi$ iff for every λ such that $q\mathcal{R}_A\lambda$, there is $i \geq 0$ such that $M, \lambda[i] \models \psi$ and for all $0 \leq j < i$ we have $M, \lambda[j] \models \varphi$.

$M, q \models \mathcal{UP}_A X\varphi$ iff for every λ such that $M, \lambda[1] \models \varphi$, we have $q\mathcal{R}_A\lambda$;
$M, q \models \mathcal{UP}_A G\varphi$ iff for every λ such that $M, \lambda[i] \models \varphi$ for all $i \geq 0$, we have $q\mathcal{R}_A\lambda$;
$M, q \models \mathcal{UP}_A \varphi\mathcal{U}\psi$ iff for every λ, such that $M, \lambda[i] \models \psi$ for some $i \geq 0$ and $M, \lambda[j] \models \varphi$ for all $0 \leq j < i$, we have $q\mathcal{R}_A\lambda$.

One of the most appealing temporal constraints is that of a deadline: some property φ should be achieved within a number (say n) of steps. This could be just expressed by $\mathcal{O}X^n\varphi$ [3]: only these courses of action are acceptable, in which the deadline is met. Note that the DATL obligation $\mathcal{O}(\langle\langle\Gamma\rangle\rangle X)^n\varphi$ expresses a different property: these are Γ who *must be able* to meet the deadline.

Fairness-like properties are also a very natural area to reason about deontic constraints. Suppose we have a resource p that can only be used by one agent at the time (and as long as a is using it, p_a is true). The constraint that every agent should always be able to use the resource is expressed by $\bigwedge_{a\in\Sigma}\mathcal{O}G\langle\langle a\rangle\rangle Gp_a$ – or, if this is an obligation of a particular scheduler s, we could write $\mathcal{O}_s$ rather than $\mathcal{O}$. Finally, let $[\![\Gamma]\!]\Phi$ be the shorthand for $\neg\langle\langle\Gamma\rangle\rangle\neg\Phi$ (coalition Γ cannot prevent φ from being the case). Then, formula $\mathcal{O}G(\langle\langle\Gamma\rangle\rangle F\varphi \rightarrow [\![\Gamma]\!]G(\varphi \rightarrow \langle\langle\Gamma'\rangle\rangle F\neg\varphi))$ says that only these courses of action are acceptable in which, might coalition Γ ever have a way to enforce φ, then it must "pass the token" to Γ' and give the other agents the ability to reverse this again.

Note also that DTATL formulas $\mathcal{UP}\psi$ express a kind of "the end justifies means" properties. For instance, $\mathcal{UP}F\mathsf{kdead}$ means that *every* course of action, which yields the killer dead, is acceptable.

2.5 Deontic ATL and Social Laws

We mentioned the two main streams in deontic logic, having either states of affairs or actions as their object of constraints. In Deontic ATL, one can express deontic requirements about *who is responsible* to achieve something, without specifying how it should be achieved. The requirement $\mathcal{O}\neg\langle\langle\{a,b\}\rangle\rangle F\mathsf{safe\text{-}open}$, for example, states that it should be impossible for a and b to bring about the disclosure of a safe in a bank. However, with c being a third employee, we might have $\mathcal{O}(\neg\langle\langle\{a,b\}\rangle\rangle F\mathsf{safe\text{-}open} \wedge \langle\langle\{a,b,c\}\rangle\rangle G\mathsf{safe\text{-}open})$: as a team of three, they *must* be able to do so! We can also express delegation, as in $\mathcal{O}_a\langle\langle b\rangle\rangle G\varphi$: authority a has the obligation that b can always bring about φ.

A recent paper [19] also addresses the issue of prescribed behavior in the context of ATL: behavioral constraints (specific model updates) are defined for ATL models, so that some objective can be satisfied in the updated model. The emphasis in [19] is on how the effectiveness, feasibility and synthesis problems in the area of social laws [18] can be posed as ATL model checking problems. One of the main questions addressed is: given a concurrent game structure M and a social law with objective φ (which we can loosely translate as $\mathcal{O}\varphi$), can we modify the original structure M into M', such that M' satisfies $\langle\langle\emptyset\rangle\rangle G\varphi$? In other words, we ask whether the overall system can be altered in such a way that it cannot but satisfy the requirements. [19] does not address the question whether certain coalitions are *able* to "act according to the law"; the law is *imposed* on the system as a whole. Thus, the approach of that paper is prescriptive, while our approach in this paper is rather descriptive. Moreover, [19] lacks explicit deontic notions in the object level.

[3] $\mathcal{O}X^n\varphi$ is not a DTATL formula, but the logic can be easily extended to include it.

An example of a requirement that cannot be imposed on the system as a whole (taken from [19]) is $p \wedge \langle\langle A\rangle\rangle X\neg p$: property p is obligatory, but at the same time, A should be able to achieve $\neg p$. This kind of constraints could be used to model "a-typical" situations, (such as: "it is obligatory that the emergency exit is not used, although at the same time people in the building should always be able to use it"). Putting such an overall constraint upon a system S means that S should both guarantee p and the possibility of deviating from it, which is impossible. It seems that our Deontic ATL covers a more local notion of obligation, in which $\mathcal{O}(p \wedge \langle\langle A\rangle\rangle X\neg p)$ can well be covered in a non-trivial way.

On the other hand, our "stability" requirements are rather weak: to demand that every obligation $\mathcal{O}\varphi$ is implementable by a coalition does not yet guarantee that the system *does* behave well. Rather, we might be looking for something in between the universal guarantee and a coalitional efficiency with respect to constraint φ. And it is one of the features of Deontic ATL – that one can express many various stability requirements, making explicit who is responsible for what.

3 Axioms, Model Checking and Similar Stories

Let ATL and DL be the languages for ATL and deontic logic, respectively, and let $\mathcal{ATL}$ and $\mathcal{DL}$ be their respective semantic structures. Then – if we do not have any mixing axioms relating the coalitional and the deontic operators – we obtain a logic $\mathsf{DATL} = \mathsf{ATL} \oplus \mathsf{DL}$ which can be called an *independent combination* of the modal logics in question [8]. [8] gives also an algorithm for model checking such a logic, given two model checkers for each separate logics. The communication overhead for combining the two model checkers would be in the order of $m + \sum_{A\in\wp(\sigma)} m_j + n \cdot l$, where m is the number of coalitional transitions in the model, m_A is the cardinality of the deontic access of coalition A, n is the number of states and l the complexity of the formula, leaving the model checking complexity of $\mathsf{ATL} \oplus \mathsf{DATL}$ linear in the size of the model and the formula [8]. However, two technical remarks are in order here. First, the formal results from [8] refer to combining temporal logics, while neither ATL nor DL is a temporal logic in the strictest sense. Moreover, the algorithm they propose for model checking of an independent combination of logics assumes that the models are finite (while there is no such assumption in our case). Nevertheless, polynomial model checking of DATL is of course possible, and we show how it can be done in Section 3.2, through a reduction of the problem to ATL model checking.

3.1 Imposing Requirements through Axioms

Following a main stream in deontic logic, we can take every deontic modality to be **KD** – the only deontic property (apart from the K-axiom and necessitation for $\mathcal{O}_\Gamma$) being the D-axiom $\neg\mathcal{O}_\Gamma\bot$. An axiomatization of ATL has been recently shown in [11]. If we do not need any mixing axioms, then the axiomatization of DATL can simply consist of the axioms for ATL, plus those of DL.

Concerning the global requirements, note that endowing $\mathcal{ATL}$ with a violation atom V is semantically very easy. Evaluating whether $\mathcal{O}\varphi$ is true at state q suggests incorporating a *universal modality* (cf. [10]) although some remarks are in place here. First of all, it seems more appropriate to use this definition of global requirements in *generated models* only, i.e., those models that are generated from some initial state q_0, by the transitions that the grand coalition Σ can make. Otherwise, the obligations might be unnecesarily weakened by considering violations or their absence in *unreachable states*. As an example, suppose we have a system that has two modes: starting from q_1, the constraint is that it is a violation to drive on the left hand side of the road ℓ, and when the system originates from q_2, one should adhere to driving on the right hand side (r). Seen as a global requirement, we would have $\mathcal{O}(\ell \vee r)$, which is of course too weak; what we want is $\mathcal{O}\ell$ (for the system rooted in q_1), or $\mathcal{O}r$ (when starting in q_2). Thus, a sound definition of obligations in a system with root q_0 is, that $M, q \models \mathcal{O}\varphi$ iff $M, q_0 \models \langle\langle\emptyset\rangle\rangle G(\neg V \rightarrow \varphi)$.

Second, we note in passing that by using the global requirement definition of obligation, the $\mathcal{O}$ modality obtained in this way is a KD45 modality, which means that we inherit the properties $\mathcal{O}\varphi \rightarrow \mathcal{O}\mathcal{O}\varphi$ and $\neg\mathcal{O}\varphi \rightarrow \mathcal{O}\neg\mathcal{O}\varphi$, as was also observed in [14]. But also, we get mixing axioms in this case: every deontic subformula can be brought to the outmost level, as illustrated by the valid scheme $\langle\langle\Gamma\rangle\rangle F\mathcal{O}\varphi \leftrightarrow \mathcal{O}\varphi$ (recall that we have $M, q \models \mathcal{O}\varphi$ iff $M, q_0 \models \mathcal{O}\varphi$ iff $M, q' \models \mathcal{O}\varphi$, for all states q, q' and root q_0). Some of the properties we have mentioned earlier in this paper can constitute interesting mixing axioms as well. For instance, a minimal property for requirements might be

$$\mathcal{O}_\Gamma\varphi \rightarrow \langle\langle\Gamma\rangle\rangle F\varphi$$

saying that every coalition can achieve its obligations. Semantically, we can pinpoint such a property as follows. Let us assume that this is an axiom scheme, and the model is distinguishing (i.e., every state in the model can be characterized by some DATL formula). Then the scheme corresponds to the semantic constraint:

$$\forall q \exists F_\Gamma \forall \lambda \in out(q, F_\Gamma) : states(\lambda) \cap img(q, \mathcal{R}_\Gamma) \neq \emptyset$$

where $states(\lambda)$ is the set of all states from λ, and $img(q, R) = \{q' \mid qRq'\}$ is the image of q with respect to relation R. In other words, Γ can enforce that every possible computation goes through at least one perfect alternative of q.

3.2 Model Checking Requirements and Abilities

In this section, we present a satisfiability preserving interpretation of DATL into ATL. The interpretation is very close to the one from [9], which in turn was inspired by [17]. The main idea is to leave the original temporal structure intact, while extending it with additional transitions to "simulate" deontic accessibility links. The simulation is achieved through new "deontic" agents: they can be passive and let the "real" agents decide upon the next transition (action *pass*), or enforce a "deontic" transition. More precisely, the "positive deontic agents" can point out a state that was deontically accessible in the original model (or, rather, a special "deontic" copy of the original state), while the "negative deontic agents" can enforce a transition to a state that was *not* accessible. The first ones are necessary to translate formulas of shape $\mathcal{O}_A\varphi$; the latter are used for the "unconditionally permitted" operator $\mathcal{UP}_A$.

As an example, let M be the deontic game structure from Figure 3, and let us consider formulas $\mathcal{O}_\Sigma\mathsf{saved}$, $\mathcal{UP}_\Sigma\mathsf{saved}$ and $\langle\langle k,b\rangle\rangle X\mathsf{pdead}$ (note that all three formulas are true in M, q_3). We construct a new concurrent game structure M^{ATL} by adding two deontic agents: $r_\Sigma, \bar{r}_\Sigma$, plus "deontic" copies of the existing states: $q_3^{\mathsf{r}_\Sigma}, q_7^{\mathsf{r}_\Sigma}, q_8^{\mathsf{r}_\Sigma}$ and $q_3^{\bar{\mathsf{r}}_\Sigma}, q_7^{\bar{\mathsf{r}}_\Sigma}, q_8^{\bar{\mathsf{r}}_\Sigma}$ (cf. Figure 4). Agent r_Σ is devised to point out all the perfect alternatives of the actual state. As state q_3 has only one perfect alternative (i.e., q_3 itself), r_Σ can enforce the next state to be $q_3^{\mathsf{r}_\Sigma}$, provided that all other relevant agents remain passive[4]. In consequence, $\mathcal{O}_\Sigma\mathsf{saved}$ translates as: $\neg\langle\langle r_\Sigma, \bar{r}_\Sigma\rangle\rangle X(\mathsf{r}_\Sigma \wedge \mathsf{saved})$. In other words, it is not possible that r_Σ points out an alternative of q_3 (while $\bar{r}_\Sigma$ obediently passes), in which saved does *not* hold.

Agent $\bar{r}_\Sigma$ can point out all the *imperfect* alternatives of the current state (for q_3, these are represented by: $q_7^{\bar{\mathsf{r}}_\Sigma}, q_8^{\bar{\mathsf{r}}_\Sigma}$). Now, $\mathcal{UP}_\Sigma\mathsf{saved}$ translates as $\neg\langle\langle r_\Sigma, \bar{r}_\Sigma\rangle\rangle X$ $(\bar{\mathsf{r}}_\mathsf{A} \wedge \mathsf{saved})$: $\bar{r}_\Sigma$ cannot point out an unacceptable state in which saved holds, hence the property of saved guarantees acceptability. Finally, $\langle\langle k,b\rangle\rangle X\mathsf{pdead}$ translates as $\langle\langle k,b,r_\Sigma,\bar{r}_\Sigma\rangle\rangle X(\mathsf{act} \wedge \mathsf{pdead})$: the strategic structure of the model has remained intact, but we must make sure that both deontic agents are passive, so that a non-deontic transition (an "action" transition) is executed.

We present the whole translation below in a more formal way. An interested reader can refer to [9] for a detailed presentation of the method, and proofs of correctness.

Given a deontic game structure $M = \langle \Sigma, Q, \Pi, \pi, Act, d, \delta, \mathbb{R}\rangle$ for a set of agents $\Sigma = \{a_1, ..., a_k\}$, we construct a concurrent game structure $M^{ATL} = \langle \Sigma', Q', \Pi', \pi', Act', d', \delta'\rangle$ in the following manner:

- $\Sigma' = \Sigma \cup \Sigma^r \cup \Sigma^{\bar{r}}$, where $\Sigma^r = \{r_A \mid A \subseteq \Sigma, A \neq \varnothing\}$ is the set of "positive", and $\Sigma^{\bar{r}} = \{\bar{r}_A \mid A \subseteq \Sigma, A \neq \varnothing\}$ is the set of "negative" deontic agents;
- $Q' = Q \cup \bigcup_{A\subseteq\Sigma, A\neq\varnothing}(Q^{\mathsf{r}_\mathsf{A}} \cup Q^{\bar{\mathsf{r}}_\mathsf{A}})$. We assume that Q and all $Q^{\mathsf{r}_\mathsf{A}}, Q^{\bar{\mathsf{r}}_\mathsf{A}}$ are pairwise disjoint. Further we will be using the more general notation $S^\mathsf{e} = \{q^\mathsf{e} \mid q \in S\}$ for any $S \subseteq Q$ and proposition e;
- $\Pi' = \Pi \cup \{\mathsf{act}, ..., \mathsf{r}_\mathsf{A}, ..., \bar{\mathsf{r}}_\mathsf{A}, ...\}$, and $\pi'(p) = \pi(p) \cup \bigcup_{A\subseteq\Sigma}(\pi(p)^{\mathsf{r}_\mathsf{A}} \cup \pi(p)^{\bar{\mathsf{r}}_\mathsf{A}})$ for every $p \in \Pi$. Moreover, $\pi'(\mathsf{act}) = Q$, $\pi'(\mathsf{r}_\mathsf{A}) = Q^{\mathsf{r}_\mathsf{A}}$, and $\pi'(\bar{\mathsf{r}}_\mathsf{A}) = Q^{\bar{\mathsf{r}}_\mathsf{A}}$;
- $d'_q(a) = d_q(a)$ for $a \in \Sigma, q \in Q$: choices of the "real" agents in the original states do not change,
- $d'_q(r_A) = \{pass\} \cup img(q, \mathcal{R}_A)^{\mathsf{r}_\mathsf{A}}$, and $d'_q(\bar{r}_A) = \{pass\} \cup (Q \setminus img(q, \mathcal{R}_A))^{\bar{\mathsf{r}}_\mathsf{A}}$. Action *pass* represents a deontic agent's choice to remain passive and let other agents choose the next state. Note that other actions of deontic agents are simply labeled with the names of deontic states they point to;

[4] We can check the last requirement by testing whether the transition leads to a deontic state of r_Σ (proposition r_Σ). It can happen only if all other relevant deontic agents choose action *pass*.

- $Act' = Act \cup \bigcup_{q \in Q, A \subseteq \Sigma}(d'_q(r_A) \cup d'_q(\bar{r}_A))$;
- the new transition function for $q \in Q$ is defined as follows (we put the choices from deontic agents in any predefined order):

$$\delta'(q, \alpha_{a_1}, ..., \alpha_{a_k}, ..., \alpha_r, ...) = \begin{cases} \delta(q, \alpha_{a_1}, ..., \alpha_{a_k}) & \text{if all } \alpha_r = pass \\ \alpha_r & \text{if } r \text{ is the first active (positive or negative) deontic agent} \end{cases}$$

- the choices and transitions for the new states are exactly the same: $d'(q^{\mathsf{r_A}}, a) = d'(q^{\bar{\mathsf{r}}_\mathsf{A}}, a) = d'(q, a)$, and $\delta'(q^{\mathsf{r_A}}, \alpha_{a_1}, ..., \alpha_{r_\Gamma}, ...) = \delta'(q^{\bar{\mathsf{r}}_\mathsf{A}}, \alpha_{a_1}, ..., \alpha_{r_\Gamma}, ...) = \delta'(q, \alpha_{a_1}, ..., \alpha_{a_k}, ..., \alpha_{r_\Gamma}, ...)$ for every $q \in Q, a \in \Sigma', \alpha_a \in d'(q, a)$.

Now, we define a translation of formulas from DATL to ATL corresponding to the above described interpretation of DATL models into ATL models:

$$\begin{aligned}
tr(p) &= p, \qquad \text{for } p \in \Pi \\
tr(\neg\varphi) &= \neg tr(\varphi) \\
tr(\varphi \vee \psi) &= tr(\varphi) \vee tr(\psi) \\
tr(\langle\langle A\rangle\rangle X\varphi) &= \langle\langle A \cup \Sigma^r \cup \Sigma^{\bar{r}}\rangle\rangle X(\mathsf{act} \wedge tr(\varphi)) \\
tr(\langle\langle A\rangle\rangle G\varphi) &= tr(\varphi) \wedge \langle\langle A \cup \Sigma^r \cup \Sigma^{\bar{r}}\rangle\rangle X \langle\langle A \cup \Sigma^r \cup \Sigma^{\bar{r}}\rangle\rangle G(\mathsf{act} \wedge tr(\varphi)) \\
tr(\langle\langle A\rangle\rangle \varphi \mathcal{U} \psi) &= tr(\psi) \vee (tr(\varphi) \wedge \langle\langle A \cup \Sigma^r \cup \Sigma^{\bar{r}}\rangle\rangle X \langle\langle A \cup \Sigma^r \cup \Sigma^{\bar{r}}\rangle\rangle \\
&\qquad (\mathsf{act} \wedge tr(\varphi))\,\mathcal{U}(\mathsf{act} \wedge tr(\psi))) \\
tr(\mathcal{O}_A\varphi) &= \neg\langle\langle \Sigma^r \cup \Sigma^{\bar{r}}\rangle\rangle X(\mathsf{r_A} \wedge \neg tr(\varphi)) \\
tr(\mathcal{UP}_A\varphi) &= \neg\langle\langle \Sigma^r \cup \Sigma^{\bar{r}}\rangle\rangle X(\bar{\mathsf{r}}_\mathsf{A} \wedge tr(\varphi)).
\end{aligned}$$

Proposition 1. *For every* DATL *formula φ, model M, and a state $q \in Q$, we have $M, q \models \varphi$ iff $M^{ATL}, q \models tr(\varphi)$.*

Proposition 2. *For every* DATL *formula φ, model M, and "action" state $q \in Q$, we have $M^{ATL}, q \models tr(\varphi)$ iff $M^{ATL}, q^{\mathsf{e}} \models tr(\varphi)$ for every* $\mathsf{e} \in \Pi' \setminus \Pi$.

Corollary 1. *For every* DATL *formula φ and model M, φ is satisfiable (resp. valid) in M iff $tr(\varphi)$ is satisfiable (resp. valid) in M^{ATL}.*

Note that the vocabulary (set of propositions Π) only increases linearly (and certainly remains finite). Moreover, for a specific DATL formula φ, we do not have to include all the deontic agents r_A and $\bar{r}_A$ in the model – only those for which $\mathcal{O}_A$ or $\mathcal{UP}_A$ occurs in φ. Also, we need deontic states only for these coalitions A. The number of such coalitions is never greater than the complexity of φ. Let m be the cardinality of the "densest" modal accessibility relation – either deontic or temporal – in M, and l the complexity of φ. Then, the "optimized" transformation gives us a model with $m' = O(lm)$ transitions, while the new formula $tr(\varphi)$ is only linearly more complex than φ [5]. In consequence, we can use the ATL model checking algorithm from [1] for an efficient model checking of DATL formulas – the complexity of such process is $O(m'l') = O(ml^2)$.

[5] The length of formulas may suffer an exponential blow-up; however, the number of *different subformulas* in the formula only increases linearly. This issue is discussed in more detail in [9].

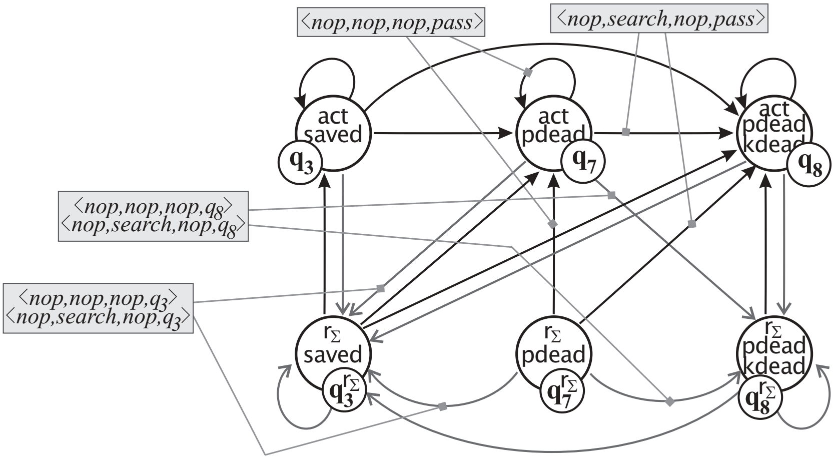

Fig. 4. ATL interpretation for the deontic game structure from Figure 3

Let us consider again the deontic game structure from Figure 3. We construct a corresponding concurrent game structure, optimized for model checking of the DATL formula $\mathcal{O}_\Sigma(\neg\mathsf{pdead} \wedge \langle\!\langle k\rangle\!\rangle X \neg\mathcal{O}_\Sigma\neg\mathsf{pdead})$: it is required that the Prime Minister is alive, but the killer is granted the ability to change this requirement. The result is shown in Figure 4. The translation of this formula is:

$$\neg\langle\!\langle r_\Sigma\rangle\!\rangle X(\mathsf{r}_\Sigma \wedge \neg(\neg\mathsf{pdead} \wedge \langle\!\langle k, r_\Sigma\rangle\!\rangle X(\mathsf{act} \wedge \neg\neg\langle\!\langle r_\Sigma\rangle\!\rangle X(\mathsf{r}_\Sigma \wedge \neg\neg\mathsf{pdead}))))$$

which holds in states q_3 and $q_3^{\mathsf{r}_\Sigma}$ of the concurrent game structure.

4 Conclusions

In this paper, we have brought obligations and abilities of agents together, enabling one to reason about what coalitions should achieve, but also to formulate principles regarding who can maintain or reinstall which ideal states or courses of action. We think the tractable model checking of DATL properties makes the approach attractive as a verification language for normative multi-agent systems.

However, as stated repeatedly in the paper, it is at the same time a report of ideas rather than of a crystallized and final analysis. We have not looked at an axiomatization of any system with non-trivial mixing axioms, nor have we yet explored some obvious routes that relate our approach in a technical sense with the work on social laws or the formal approaches that enrich ATL with an epistemic flavor, for instance. Nevertheless, we believe we have put to the force the fact that indeed DATL is a very attractive framework to incorporate abilities of agents and teams with deontic notions. We hope that the growing community, interested in norms in the computational context, can provide some feedback to help making appropriate decisions in the many design choices that we left open.

References

1. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002. Updated, improved, and extended text. Available at `http://www.cis.upenn.edu/~alur/Jacm02.pdf`.
2. A.R. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103, 1958.
3. N. Belnap. Backwards and forwards in the modal logic of agency. *Philosophy and Phenomenological Research*, LI(4):777–807, 1991.
4. N. Belnap and M. Perloff. Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199, 1988.
5. P. Bieber and F. Cuppens. Expression of confidentiality policies with deontic logics. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 103–123. John Wiley & Sons, 1993.
6. J. Broersen, M. Dastani, Z. Huang, and L. van der Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 9–16, 2001.
7. J. Fiadeiro and T. Maibaum. Temporal reasoning over deontic specifications. *Journal of Logic and Computation*, 1(3):357–396, 1991.
8. M. Franceschet, A. Montanari, and M. de Rijke. Model checking for combined logics. In *Proceedings of ICTL*, 2000.
9. V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, section on *Knowledge, Rationality and Action*, 2004. To appear.
10. V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2(1):5–30, 1992.
11. V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of the alternating-time temporal logic. Submitted, 2003.
12. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
13. H.J. Levesque. All I know: a study in auto-epistemic logic. *Artificial Intelligence*, 42(3):263–309, 1990.
14. A. Lomuscio and M. Sergot. Deontic interpreted systems. *Studia Logica*, 75(1):63–92, 2003.
15. J.-J.Ch. Meyer. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame Journal of Formal Logic*, 29(1):109–136, 1988.
16. J.-J.Ch. Meyer and R.J. Wieringa. Deontic logic: A concise overview. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 3–16. John Wiley & Sons, 1993.
17. K. Schild. On the relationship between BDI logics and standard logics of concurrency. *Autonomous Agents and Multi Agent Systems*, pages 259–283, 2000.
18. Y. Shoham and M. Tennenholz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of AAAI-92*, 1992.
19. W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility and synthesis. Submitted, 2004.
20. J. van Eck. A system of temporally relative modal and deontic predicate logic and its philosophical applications. *Logique et Analyse*, 100:249–381, 1982.
21. G.H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.
22. R.J. Wieringa and J.-J.Ch. Meyer. Applications of deontic logic in computer science: A concise overview. In J.-J.Ch. Meyer and R.J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 17–40. 1993.