Formal Verication of Ethical Properties in Multiagent Systems

Bruno Mermet, Gaële Simon

Formal Verification of Ethical Properties in Multiagent Systems

Bruno Mermet and **Gaële Simon**[1]

Abstract. The increasing use of autonomous artificial agents in hospitals or in transport control systems leads to consider whether moral rules shared by many of us are followed by these agents. This is a particularly hard problem because most of these moral rules are often not compatible. In such cases, humans usually follow ethical rules to promote one moral rule or another. Using formal verification to ensure that an agent follows a given ethical rule could help in increasing the confidence in artificial agents. In this article, we show how a set of formal properties can be obtained from an ethical rule ordering conflicting moral rules. If the behaviour of an agent entails these properties (which can be proven using our existing proof framework), it means that this agent follows this ethical rule.

1 Introduction

The introduction of autonomous artificial agents in domains like health or high-frequency trading could lead to numerous problems if these agents are not able to understand and to take into account moral rules. For example, agents able to understand and to use the code of medical ethics could base their decision on ethical motivations in order to choose which piece of information must be provided, according to the medical confidentiality. This explains that the research community [13, 23] seems recently to focus on ethics for autonomous agents, which lead to numerous articles [4, 18, 22] and conferences[2].

This article takes place in ETHICAA project[3] which aims at dealing with management of moral and ethical conflicts between autonomous agents. If existing works are mainly focused on ethical decision and reasoning questions, [5, 27, 30, 2, 8, 16], there are very few proposals dedicated to formal verification of such behaviours. But the main specificity of moral and ethical codes is that, acccording to the context, they may be not entailed by agents or by people and it must be considered as a normal situation. For example, in a human context, if stealing is not considered as a moral action, somebody stealing because of hunger is not considered as immoral.

As a consequence, this article presents a work which aims at proposing a framework for the formal specification and the formal verification of the behaviour of an autonomous agent from an ethical point of view. As stated in the work of Abramson and Pike, a moral rule is represented by a formal property that must be entailed by an agent [1]. As a consequence, the behaviour of an agent is an ethical one if it entails all the expected moral rules in a given context.

Considering that a moral rule can be represented by a first order logical formula $\mathcal{F}$ with enough expressiveness for most practical cases, our goal is to establish that the behaviour of an agent is an ethical one if it entails $\mathcal{F}$. If not, then the behaviour is not an ethical one. However, such a logical system is only semi-decidable: it is not always possible to prove that a system does not entail a formula $\mathcal{F}$. Indeed, if an automatic prover does not manage to prove that the behaviour of an agent entails a formula $\mathcal{F}$, it is not possible to automatically determine if it results from the fact that the behaviour does not actually entail $\mathcal{F}$ or if it is because the prover can not prove the opposite.

As a consequence, we propose to use a formal framework allowing to reduce as far as possible the number of correct formulae that can not automatically be proven. In section 2, such formal frameworks are described, especially those dedicated to multiagent systems. Then what we call moral and ethical rules are defined. In section 3, our formal system is described ans its use in an ethical context is presented in section 4.

2 Śtate of the art

Since the early days of computing, the need to ensure the correctness of softwares is a major issue for software developers. This need has become crucial with critical systems, that is to say applications dedicated to domains where safety is vital (as transport for example). However, formally proving a software is a long and difficult process which can conflict with profitability and efficiency criteria of some companies. There are two main kinds of validation processes: test and proof. In this article, we only focus on the second one. Proofs can be performed either by model checkers, or by theorem provers. Model-checkers are basically based on an exhaustive test principle whereas theorem provers often use sequent calculus and heuristics in order to generate proofs.

Even if the proof process can be a long and difficult one, it allows to prove very early specifications which can then be refined progressively until an executable code is obtained with proofs at each step. So errors are detected early in the process which reduces their cost. Refinement allows also to simplify formulae to prove at each step enabling their automatic proof. These proofs are based on a formal specification expressed thanks to a formal language.

[1] Laboratoire GREYC - UMR 6072, Université du Havre, France, email:Bruno.Mermet@unicaen.fr

[2] Symposium on Roboethics, International Conference on Computer Ethics and Philosophical Enquiry, Workshop on AI and Ethics, International Conference on AI and Ethics.

[3] http://ethicaa.org/

2.1 Models and methods dedicated to MAS

The main goal of models dedicated to MAS is to help developers to design multiagent systems. A lot of models have been proposed, but the most well-known of them is surely the BDI model [26] which has become a standard with several extensions.

MetateM [15] and Desire [7] are among the first proposed formal methods dedicated to MAS. However, they don't allow to specify properties that are expected to be verified by the system.

In Gaia [32], a MAS is specified twice: as a first step, its behaviour is specified especially thanks to safety properties and then invariant properties are introduced. Thus, this method proposes foundations for proving properties about agents behaviour. However, such proofs are not really possible with Gaia because properties are not associated to agents but to roles, and there is no formal semantics specifying how the different roles must be combined.

There is another kind of method: goal-oriented methods. Most of them, however, are dedicated to agents specification, and, seldom, provide tools for the system level which implies that the agentification phase must have been achieved before. Two exceptions can be mentioned: Moise [17] and PASSI [10]. For example, as far as PASSI is concerned, agent types are built gathering *use cases* identified during the analysis phase. However, there is no guidelines for the gathering process.

Finally, more recently, Dastani *et al.* have proposed the 2APL language [11]. Unfortunately, this formal language does not include any proof system. Moreover, 2APL is not compositional which leads to a too much monolithic system in a proof context.

2.2 Models and methods dedicated to proof

As stated before, there are mainly two approaches to check if a specification is correct: model-checking and theorem-proving.

Most of works in this area dedicated to agents use model-checking [6, 25]. however, all these proposals share the same limit: the combinatorial explosion of the possible system executions makes the proof of complex MAS very difficult. As a matter of fact, these approaches often reduce the proof to propositional formulae and not predicates.

Works dealing with the use of theorem proving for MAS proof are quite unusual. It is certainly because, first-order logic being only semi-decidable, proof attempts must be achieved using heuristics and the proof of a true property can fail. However, now, numerous theorem provers, like PVS [24], are able to prove automatically complex properties.

There are other models based on logic programming, such as CaseLP and DCaseLP [3] which are most suited to theorem proving than the previous one. But, it seems that only proofs on interaction protocols can be performed using these models.

Congolog [12] and CASL [28] are also two interesting languages, based on situation calculus. Moreover, they allow to perform proofs. But these proofs are focused only on actions sequencing. It is not possible to reason about their semantics.

2.3 Ethical and moral rules

Both in philosophy and in latest research in neurology and in cognitive sciences, concepts like moral and ethics have been discussed. Although these words initially mean the same thing, a distinction between them has been introduced by some authors [9, 29]. Indeed, moral establishes rules allowing to evaluate situations or actions as good or bad. Ethics allows to reconcile moral rules when a conflict occurs or when there are difficulties in their application. In the work presented in this paper, our definitions are based on this distinction.

2.3.1 Moral rules

In our point of view, a moral rule is a rule describing, in a given context, which states of the system must be considered as good or bad. Thus, moral rules can be specified by a formula like $context \rightarrow P_{var}$, with P_{var} being a predicate defined on variables known by agents. So, a moral rule can be seen as a specific conditional invariant property in that it is not necessary to check it in order to ensure a correct execution of the system. But it must be established if the agent must be used in a system in which the rule is expected to be entailed. For example, in the context of an autonomous car, the property $lane = highway \rightarrow speed \leq 130$ can be considered as a safety property. As a consequence, this property must be fulfilled by the agent behaviour. Nevertheless, in order to avoid life-threatening, a caution moral rule r_p states that, when there is ice on the road, the car can not have a speed greater than 30 km/h. Formally, this rule is specified as: $weather = ice \rightarrow speed \leq 30$. This property needs not to be entailed in order for the car to have a valid behaviour in general. But it must be taken into account in systems in which preservation of life is considered.

2.3.2 Ethical rules

When an individual or an agent follows several moral rules, it sometimes happens that two rules, or more, enter in conflict with one another. In such a situation, an ethical rule specifies what should be done. If some rules like the doctrine of double-effect [19] can be complex ones, we consider in our work that an ethical rule is a rule stating, in a conflict situation, the sequence in which the moral rules should be adressed by the agent. We also consider that an ethical rule is contextual: it may lead to different decisions according to the circumstances. Considering the autonomous car example, in order to respect other drivers, a moral rule r_r can be introduced. This new rule states that, when driving on a highway, the speed can not be lower than 80 km/h which can be formally specified as $lane = highway \rightarrow speed \geq 80$. This rule may conflict with the r_p rule described before: if there is ice on the road and if the car uses an highway, according to r_p, its speed must be lower than 30 km/h but is must also be greater than 80 km/h according to r_r. An ethical rule can, for example, states that, in any case, caution (specified by r_p) must be preferred to respect (specified by r_r). An other ethical rule could state that this preference is to be considered only in case of surgery and, in other situations, that the preference must be inverted.

2.4 Very little works about verification of ethics

Dealing with ethical problems with formal approaches is studied especially in [1]. In this article, authors explain why using formal approaches could be interesting to ensure that agents fulfill ethical rules. However it is only a position paper: there is no proposed concrete method to implement these principles.

In [14], authors propose to specify and to formally prove the ethical decision process described in [31]: when a choice between different actions must be made, a value is associated to each possible action according to the safety level provided by the action. As a consequence, if an action A is considered to be safer than an other one, then A is executed. There is yet a major drawback to this approach: the ethical dimension is taken into account only during a choice between actions which must be managed using the decision procedure described before. Thus, this work is not general enough to provide an effective handling of ethics.

3 GDT4MAS

To take ethical problems into account, we have decided to use the GDT4MAS approach [20, 21]. Indeed, this method, that also includes a model, exhibits several characteristics which are interesting to deal with ethical problems:

- This method proposes a formal language to specifiy not only properties an agent or a MAS must entail but also the behaviour of agents;
- Properties are specified using first-order logic, a well-known and expressive formal notation;
- The proof process can be managed automatically.

In next sections, the GDT4MAS method is summarized. More details can be found in previous cited articles.

3.1 Main concepts

Using GDT4MAS requires to specify 3 concepts: the environment, agent types and agents themselves which are considered as instances of agent types. In the remainder of this section, each of these parts is briefly described.

Environment The environment is specified by a set of typed variables and by an invariant property $i\varepsilon$.

Agents types Agent types are specified each by a set of typed variables, an invariant property and a behaviour. An agent behaviour is mainly defined by a *Goal Decomposition Tree* (GDT). A GDT is a tree where each node is a goal. Its root node is associated to the main goal of the agent. A plan, specified by a sub-tree, is associated to each goal: when this plan is successfully executed, it means that the goal associated to its root node is achieved. A plan can be made of either a simple action or a set of goals linked by a *a decomposition operator*. The reader is invited to read [20, 21] to know how goals are formally described.

Agents Agents are specified as instances of agents types, with effective values associated to agents types parameters.

3.2 GDT example

Figure 1 shows an example of GDT. The goal of the behaviour specified by this GDT is to turn on the light in a room n (n is a GDT parameter). To achieve this goal, the agent tries to enter the room. Indeed, a photoelectric cell is expected to detect when someone tries to enter the room and, then, to switch on the light. So this seems to be a relevant plan. However, the photoelectric cell does not always work properly (thus, the resolution of the goal *Entering the room* may fail) and the agent can have to use the switch. More details can be found in [21].

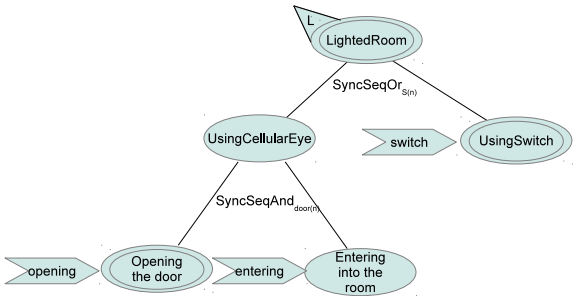

Figure 1. Example of a GDT

3.3 Proof principles

The goal of the proof mechanism proposed in GDT4MAS is to prove the following properties:

- During their execution, agents maintain their invariant property. This kind of properties states that the agent must stay in valid states;
- The behaviour of agents is sound (i.e. plans associated to goals are correct);
- Agents fullfill their liveness properties. These properties specify dynamic characteristics which must be exhibited by the agent behaviour.

Moreover, the proof mechanism is based on some key principles. Especially, *proof obligations* (ie. properties that must be proven to ensure the system correctness) can be generated automatically from a GDT4MAS specification. They are expressed in first-order logic and can be proven by any suited theorem prover. Last but not least, the proof system is a compositional one: proving the correctness of an agent consists in proving several small independant proof obligations.

4 Proving an ethics

4.1 Problem characterisation

Let consider an agent ag whose behaviour has been formally specified and whose correctness has been proven with respect to previously described properties. Let suppose that this agent must be used in a world with an ethical rule based on a set of moral rules. The question we are interested in is the following: does the behaviour of ag entails the ethical rule er ?

As GDT4MAS allows especially to prove invariant properties, we propose that moral rules and ethical rules are expressed as such properties. Indeed, most moral rules can easily be specified by invariant properties. As a consequence, we propose to structure each moral rule as:

$$\{(when_i, \{(var_i, set_i)\})\}$$

This means that each rule constrains, in different contexts, the set of values (set_i) which can be assigned to different variables (var_i). So, the caution rule r_p described in section 2.3 could be formalized as follows:

$$\{(weather = ice, \{(speed, \{0 \dots 30\})\})\}$$

However, specifying ethical rules as invariant properties is not as obvious as it is for moral rules. Indeed, they do not characterize system states but provide prioritisations on moral rules with respect to different contexts.

Let MR be the set of moral rules and let $\mathcal{P}$ be the set of predicates on variables which can be perceived by a given agent. An ethical rule er is defined as:

$$er \in \mathcal{P} \rightarrow\!\!\!\!\!+ \; (1 \ldots card(MR) \rightarrowtail\!\!\!\!\!\!\gg MR)$$

Here, $X \rightarrow\!\!\!\!\!+ \; Y$ is the set of partial functions from X to Y and $X \rightarrowtail\!\!\!\!\!\!\gg Y$ is the set of bijections from X to Y. Therefore, informally, this definition means that, in some cases characterized by a given predicate $p \in \mathcal{P}$, moral rule MR are prioritized. For example, if $p \in \mathcal{P}$ is a predicate, $er(p)(1)$ defines the moral rule with the highest priority when p is true, $er(p)(2)$ defines the one with the second highest priority and so on.

To examplify this principle, here is an example: an agent A_1 must choose the color of a traffic light $tl1$ which stands on road $r1$, at a crossroad with road $r2$. In the system in which this agent acts, two moral rules stand. The first one states that, to avoid accidents, when the traffic light on road $r2$ is *green* or *orange* then $tl1$ can not be *green*. This rule can be formalized as:

$$\{(tl2 \in \{green, orange\}, \{tl1, \{orange, red\}\})\}$$

The second moral rule $mr2$ states that the road $r1$ is a very high priority road and thus, the traffic light on road $tl1$ must always be *green*. This rule can be formalized as:

$$\{(true, \{tl1, \{green\}\})\}$$

Obviously, these two rules can not be always satisfied in the same time, especially when the second traffic light is *green*. In this situation, according to $mr1$, $tl1$ must be *orange* or *red* but, according to $mr2$, $tl1$ must be *green*.

Let now suppose that, in the considered system, an ethical rule er provides priorities on moral rules. For example, er states that $r1$ is a priority road unless $tl2$ is *green* or *orange*. In other words, this means that $mr1$ has always a higher priority than $mr2$. Formally, it can be expressed by:

$$\{(true, \{(1, mr1), (2, mr2)\})\}$$

4.2 Proposed solution

As part of our work, we wish to prove that the behaviour of an agent is correct with respect to an ethical rule defined on the basis of several moral rules. The behaviour of an agent can not fulfill the set of all moral rules that are relevant to it since, as explained previously, these rules may be conflicting. As a consequence, in order to ensure that the behaviour of an agent is correct with respect to a given ethical rule, we propose a *predicates transformation system* that turns predicates associated to moral rules into other predicates which can be proven, according to the priorities introduced by the ethical rule. In the work presented here, situations with only two moral rules involved are considered. But the proposed principle could be used for a system with more moral rules. The main principle is that moral rules and ethical rules are turned into a set of invariant properties, properties which can be proven with our proof system.

In the remainder, the transformation is shown in a case where only one variable is affected by moral ryles. In the general case, the same formulae must be generated for each variable appearing in the set of moral rules. If a variable appears only in a subset of moral rules, it is added in other moral rules with a unique constraint: its value must be in the variable definition domain).

Let's now consider a variable V. Let also suppose that the moral rule mr provides the following constraints on V:

$$mr1 = \left\{ \begin{array}{l} (when_{mr1_1}, (V, set_{mr1_1})) \\ (when_{mr1_2}, (V, set_{mr1_2})) \end{array} \right\}$$

Let suppose that a second moral rule $mr2$ provides the following constraints on V:

$$mr2 = \left\{ \begin{array}{l} (when_{mr2_1}, (V, set_{mr2_1})) \\ (when_{mr2_2}, (V, set_{mr2_2})) \\ (when_{mr2_3}, (V, set_{mr2_3})) \end{array} \right\}$$

Last but not least, it is also supposed that an ethical rule specifies that if the condition $cond_1$ is true, $mr1$ has the highest priority against $mr2$ and it is the opposite if the condition $cond_2$ is true. This ethical rule er is defined as follows:

$$er = \left\{ \begin{array}{l} (cond_1, \{(1, mr1), (2, mr2)\}) \\ (cond_2, \{(1, mr2), (2, mr1)\}) \end{array} \right\}$$

We can then generate a set of provable invariant properties associated to the ethical rule and to moral rules. First of all, according to er, when $cond_1$ is true, $mr1$ takes precedence:

$$cond_1 \rightarrow (when_{mr1_1} \rightarrow V \in set_{mr1_1})$$
$$cond_1 \rightarrow (when_{mr1_2} \rightarrow V \in set_{mr1_2})$$

Secondly, when $cond_1$ is true and when $mr1$ does not apply, $mr2$ must be fulfilled:

$$cond_1 \rightarrow \left(\begin{array}{l} (\neg when_{mr1_1} \wedge \neg when_{mr1_2}) \\ \rightarrow \\ (when_{mr2_1} \rightarrow V \in set_{mr2_1}) \end{array} \right)$$
$$cond_1 \rightarrow \left(\begin{array}{l} (\neg when_{mr1_1} \wedge \neg when_{mr1_2}) \\ \rightarrow \\ (when_{mr2_2} \rightarrow V \in set_{mr2_2}) \end{array} \right)$$
$$cond_1 \rightarrow \left(\begin{array}{l} (\neg when_{mr1_1} \wedge \neg when_{mr1_2}) \\ \rightarrow \\ (when_{mr2_3} \rightarrow V \in set_{mr2_3}) \end{array} \right)$$

Finally, when $cond_1$ is true and when mr_1 and mr_2 apply, if possible, a value entailing the two moral rules must be chosen:

$$\left(\begin{array}{l} (cond_1 \wedge when_{mr1_1} \wedge when_{mr2_1}) \\ \rightarrow \\ (set_{mr1_1} \cap set_{mr2_1} \neq \emptyset \rightarrow V \in set_{mr1_1} \cap set_{mr2_1}) \end{array} \right)$$
$$\left(\begin{array}{l} (cond_1 \wedge when_{mr1_1} \wedge when_{mr2_2}) \\ \rightarrow \\ (set_{mr1_1} \cap set_{mr2_2} \neq \emptyset \rightarrow V \in set_{mr1_1} \cap set_{mr2_2}) \end{array} \right)$$
$$\left(\begin{array}{l} (cond_1 \wedge when_{mr1_1} \wedge when_{mr2_3}) \\ \rightarrow \\ (set_{mr1_1} \cap set_{mr2_3} \neq \emptyset \rightarrow V \in set_{mr1_1} \cap set_{mr2_3}) \end{array} \right)$$
$$\left(\begin{array}{l} (cond_1 \wedge when_{mr1_2} \wedge when_{mr2_1}) \\ \rightarrow \\ (set_{mr1_2} \cap set_{mr2_1} \neq \emptyset \rightarrow V \in set_{mr1_2} \cap set_{mr2_1}) \end{array} \right)$$
$$\left(\begin{array}{l} (cond_1 \wedge when_{mr1_2} \wedge when_{mr2_2}) \\ \rightarrow \\ (set_{mr1_2} \cap set_{mr2_2} \neq \emptyset \rightarrow V \in set_{mr1_2} \cap set_{mr2_2}) \end{array} \right)$$
$$\left(\begin{array}{l} (cond_1 \wedge when_{mr1_2} \wedge when_{mr2_3}) \\ \rightarrow \\ (set_{mr1_2} \cap set_{mr2_3} \neq \emptyset \rightarrow V \in set_{mr1_2} \cap set_{mr2_3}) \end{array} \right)$$

Similar invariant properties must also be generated when $cond_2$ is true, but this time with $mr2$ being the moral rule with the highest priority.

Let us now use this mechanism for the previously presented example. As $cond_1$ is true, formulae can be simplified a first time. Moreover, as there is only one case (a single *when*) for $mr1$ and $mr2$, previous formulae can be simplified a second time as described in the following. When $cond_1$ is true, $mr1$ is the rule with the highest priority:

$$when_{mr1_1} \rightarrow V \in set_{mr1_1}$$

When $cond_1$ is true and when $mr1$ does not apply, $mr2$ must be taken into account:

$$(\neg when_{mr1_1}) \rightarrow (when_{mr2_1} \rightarrow V \in set_{mr2_1})$$

When $cond_1$ is true and when mr_1 and mr_2 apply, if possible, a value entailing the two moral rules must be chosen:

$$(when_{mr1_1} \wedge when_{mr2_1}) \rightarrow \left(\begin{array}{l} set_{mr1_1} \cap set_{mr2_1} \neq \emptyset \\ \rightarrow \\ V \in set_{mr1_1} \cap set_{mr2_1} \end{array} \right)$$

Moreover, the following formulae stand:

$$\left\{ \begin{array}{l} V \equiv TL1 \\ when_{mr1_1} \equiv (TL2 \in \{green, orange\}) \\ set_{mr1_1} \equiv (\{orange, red\}) \\ when_{mr2_1} \equiv (true) \\ set_{mr2_1} \equiv (\{green\}) \end{array} \right.$$

As a consequence, the following invariant property can be obtained. It must be proven in order to ensure that the behaviour of agent $a1$ is executed with respect to the ethical rule which specifies that the road $r1$ is a priority road unless the traffic light $TL2$ is *green* or *orange*:

$$TL2 \in \{green, orange\} \rightarrow TL1 \in \{orange, red\}$$
$$TL2 \notin \{green, orange\}) \rightarrow TL1 \in \{green\}$$
$$(TL2 \in \{green, orange\}) \rightarrow \left(\begin{array}{l} \{orange, red\} \cap \{green\} \neq \emptyset \\ \rightarrow \\ TL1 \in \{orange, red\} \cap \{green\} \end{array} \right)$$

As $\{orange, red\} \cap \{green\} = \emptyset$, this invariant property can be simplified:

$$TL2 \in \{green, orange\} \rightarrow TL1 \in \{orange, red\}$$
$$TL2 \notin \{green, orange\}) \rightarrow TL1 \in \{green\}$$

Therefore, thanks to the proposed predicates transformation system, a new invariant property is generated which will be maintained by any agent whose behaviour fulfills the different moral rules as specified by the ethical rule defined in the system. The proof system associated to GDT4MAS allows to prove that the formal specification of such an agent leads to a behaviour that maintains the invariant property.

4.3 Case study

In this section, an application of principles presented in the previous section to a new case study is shown. This case study is based on a more usual ethical question and has not been designed especially as a use case for our framework. It involves three agents A, B and C which have to find a meeting date. An agent can propose a date and the two other agents must inform all agents if the date suits them or not. For exemple, A can propose a date to B and C. If B or C does not accept the date, it must give a reason for its denial to other agents. Let suppose that d is a date proposed by A. C has to act with respect to the following moral rules:

- $mr1$: C does not want to hurt anybody;
- $mr2$: C must inform A and B about the reason why the date d does not suit him.

However, if the true reason that explains why C does not accept the date d can hurt A (for example a date with A's wife), the two rules $mr1$ and $mr2$ are conflicting. To manage this conflict, C is supposed to use an ethical rule er which states that, in any case, it is better not to hurt than to tell the truth.

In order to formalise this problem, some notations must be introduced. The set of answers that can be given by C to A or B is called E_{RP} and is defined as $E_{RP} = \{r1, r2, r3, r4\}$. The variable that contains the true reason which explains the denial of C is call V_{RC}. To clarify the issue, here is an example of what could be the different reasons used by C:

- $r1$: I have a date with A's wife;
- $r2$: I am sick ;
- $r3$: A is not good at organising meetings;
- $r4$: I had an accident with the car that B lended to me.

Moreover, the set of hurting answers for each agent is specified by a function $F_{RD} \in agents \rightarrow \mathcal{P}(E_{RP})$. In the example, $F_{RD} = \{(A, \{r1, r3\}), (B, \{r4\})\}$ which means that $r1$ and $r3$ are hurting answers for agent A and $r4$ is a hurting answer for agent B. The variable containing the answer to agent A is called V_{RFA} and the variable containing the answer to agent B is called V_{RFB}.

In this example, two moral rules are identified:

- $mr1$: C does not want to hurt A or B that is why its answers must be chosen among non hurting ones ;
- $mr2$: C does not want to lie that is why its answers must be true reasons.

These rules can be formalised as:

$$mr1 : \left\{ true, \left\{ \begin{array}{l} (V_{RFA}, E_{RP} - F_{RD}(A)) \\ (V_{RFB}, E_{RP} - F_{RD}(B)) \end{array} \right\} \right\}$$

$$mr2 : \{true, \{(V_{RFA}, \{V_{RC}\}), (V_{RFB}, \{V_{RC}\})\}\}$$

Finally, an ethical rule er states that, in any case, $mr1$ has a highest priority than $mr2$ which can be formalised by:

$$er = \{(true, \{(1, mr1), (2, mr2)\})\}$$

Applying principles described in the previous section, we have to add formulae given below to the invariant property associated to C (here are only shown formulae generated for V_{RFA}; similar formulae for V_{RFB} must be also added). For each formula, we summarize informally what it specifies.

When $cond_1$ is true, $mr1$ has the highest priority:

$$true \rightarrow (true \rightarrow V_{RFA} \in E_{RP} - F_{RD}(A))$$

When $cond_1$ is true, when $mr1$ does not apply, $mr2$ must be used:

$$true \rightarrow ((\neg true \wedge \neg true) \rightarrow (true \rightarrow V_{RFA} \in \{V_{RC}\}))$$

When $cond_1$ is true, when $mr1$ and $mr2$ apply, if possible, a value entailing the two moral rules must be chosen:

$$\begin{array}{l} (true \wedge true \wedge true \rightarrow \\ \quad ((E_{RP} - F_R D(A)) \cap \{V_{RC}\} \neq \emptyset \rightarrow \\ \quad\quad V_{RFA} \in (E_{RP} - F_{RD}(A)) \cap \{V_{RC}\})) \end{array}$$

This can then be simplified as follows:

$$\begin{array}{l} V_{RFA} \in E_{RP} - F_{RD}(A) \\ ((E_{RP} - F_{RD}(A)) \cap \{V_{RC}\} \neq \emptyset \rightarrow \\ \quad V_{RFA} \in (E_{RP} - F_{RD}(A)) \cap \{V_{RC}\})) \end{array}$$

If the final invariant property, obtained by adding this set of properties to the initial invariant property of agent C, is maintained by C, it ensures that the behaviour of C entails the ethical rule introduced before. And the proof system associated to GDT4MAS allows to prove that the behaviour of an agent maintains an invariant property.

As a consequence, according to these properties, in the presented case study, and in a context where the true reason for C to deny the date is $r1$, an agent whose behaviour is executed with respect to the ethical rule er should have only to ensure:

$$V_{RFA} \in \{r1, r2, r3, r4\} - \{r1, r3\}$$

This can be simplified as:

$$V_{RFA} \in \{r2, r4\}$$

On the other hand, if the true reason is $r2$, the behaviour of C should entail the two following properties:

$$V_{RFA} \in \{r1, r2, r3, r4\} - \{r1, r3\}$$
$$V_{RFA} \in (\{r1, r2, r3, r4\} - \{r1, r3\}) \cap \{r2\}$$

This implies that the only solution is: $V_{RFA} = r2$. Proceeding like that for each possible reason that can be given by C, the following table can be obtained:

V_{RC}	$r1$	$r2$	$r3$	$r4$
V_{RFA}	$r2, r4$	$r2$	$r2, r4$	$r4$

This little analysis allows to generate simple properties which must be entailed by an agent that prefers to lie than to hurt but that, when possible, tells the truth. Indeed, when the true reason does not hurt A ($r2$ or $r4$), an agent whose behaviour is proven to be correct, must have to give this reason. However, when the true reason hurts A ($r1$ or $r3$), an agent with a correct behaviour must have to lie by giving to A an other reason (here, $r2$ or $r4$).

5 Conclusion and future works

In this article, we have shown it is possible to formally prove that an agent acts with respect to potentially conflicting moral rules if there exists an ethical rule allowing to manage conflicts. Indeed, this rule must specify, when at least two moral rules are conflicting and in different contexts, priorities between the different moral rules. In order to achieve this, we have introduced predicate transformers which enable us to generate a set of consistent predicates from nonetheless conflicting moral rules. After a first simple example used to introduce concepts, we have shown with a more concrete case study that the proposed framework may be used for more real-world cases.

Other case studies are however required to really validate the scope of the proposed framework. In particular, moral rules have been restricted to rules that can be specified as disjoint assignment constraints on variables values. It seems important to evaluate the consequences of this restriction. For cases where this restriction would invalidate the proposed approach, we have to study how this framework could be extended to linked variables assignments. For example, one could imagine that the caution rule, associated to the case of driving on ice, may establish a link between the maximum speed and the angle of the car to the straigth direction as follows: $weather = ice \rightarrow speed + angle/2 \leq 40$. Indeed, the sharper is the turn taken by the car, the lower must be the speed to avoid the car to skid.

Last but not least, from a philosophical point of view, our approach must be extended in order to capture more precisely moral and ethics, especially by integrating value notion. Indeed, moral rules are generally based on values such as generosity, equality, love of the truth... and, in a specific context, ethical judgement uses a hierarchy between these values. Formally specifying the value notion is then the next step of our work.

REFERENCES

[1] D. Abramson and L. Pike, 'When formal Systems Kill: Computer Ethics and Formal Methods', *APA Newsletter on Philosophy and Computers*, **11**(1), (2011).

[2] R. Arkin, *Governing Lethal Behavior in Autonomous Robots*, Chapman and Hall, 2009.

[3] M. Baldoni, C. Baroglio, I. Gungui, A. Martelli, M. Martelli, V. Mascardi nd V. Patti, and C. Schifanella, 'Reasoning About Agents' Interaction Protocols Inside DCaseLP', in *LNCS*, volume 3476, pp. 112–131, (2005).

[4] A.F. Beavers, 'Moral machines and the threat of ethical nihilism', in *Robot ethics: the ethical and social implication of robotics*, 333–386, MIT Press, (2011).

[5] F. Berreby, G. Bourgne, and J.-G. Ganascia, 'Modelling moral reasoning and ethical responsibility with logic programming', in *20th LPAR*, pp. 532–548, (2015).

[6] R.H. Bordini, M. Fisher, W. Visser, and M. Wooldridge, 'Verifiable multi-agent programs', in *1st ProMAS*, (2003).

[7] F.M.T. Brazier, P.A.T. van Eck, and J. Treur, *Simulating Social Phenomena*, volume 456, chapter Modelling a Society of Simple Agents: from Conceptual Specification to Experimentation, pp 103–109, LNEMS, 1997.

[8] H. Coelho and A.C. da Rocha Costa. On the intelligence of moral agency, 2009.

[9] A. Comte-Sponville, *La philosophie*, PUF, 2012.

[10] M. Cossentino and C. Potts, 'A CASE tool supported methodology for the design of multi-agent systems', in *SERP*, (2002).

[11] M. Dastani, '2APL: a practical agent programming language', *JAAMAS*, **16**, 214–248, (2008).

[12] G. de Giacomo, Y. Lesperance, and H. J. Levesque, 'Congolog, a concurrent programming language based on the situation calculus', *Artificial Intelligence*, **121**(1-2), 109–169, (2000).

[13] Commission de réflexion sur l'Éthique de la Recherche en science et technologies du Numérique d'Allistene, 'éthique de la recherche en robotique', Technical report, CERNA, (2014).

[14] L.A. Dennis, M. Fisher, and A.F.T. Winfield, 'Towards Verifiably Ethical Robot Behaviour', in *Artifial Intelligence and Ethics AAAI Workshop*, (2015).

[15] M. Fisher, 'A survey of concurrent METATEM – the language and its applications', in *1st ICTL*, pp. 480–505, (1994).

[16] J.G. Ganascia, 'Modeling ethical rules of lying with answer set programming', *Ethics and Information Technologyn*, **9**, 39–47, (2007).

[17] J.F. Hubner, J.S. Sichman, and O. Boissier, 'Spécification structurelle, fonctionnelle et déontique d'organisations dans les SMA', in *JFIADSM*. Hermes, (2002).

[18] D. McDermott, 'Why ethics is a high hurdle for ai', in *North American Conference on Computers and Philosophy*, (2008).

[19] A. McIntyre, 'Doctrine of double effect', in *The Stanford Encyclopedia of Philosophy*, ed., Edward N. Zalta, winter edn., (2014).

[20] B. Mermet and G. Simon, 'Specifying recursive agents with GDTs.', *JAAMAS*, **23**(2), 273–301, (2011).

[21] B. Mermet and G. Simon, 'A new proof system to verify GDT agents.', in *IDC*, volume 511 of *Studies in Computational Intelligence*, pp. 181–187. Springer, (2013).

[22] J.H. Moor, 'The nature, importance, and difficulty of machine ethics', *IEEE Intelligent Systems*, **21**(4), 29–37, (2006).

[23] Future of Life Institute. Research priorities for robust and beneficial artificial intelligence, 2015.

[24] S. Owre, N. Shankar, and J. Rushby, 'Pvs: A prototype verification system', in *11th CADE*, (1992).

[25] F. Raimondi and A. Lomuscio, 'Verification of multiagent systems via orderd binary decision diagrams: an algorithm and its implementation', in *3rd AAMAS*, (2004).

[26] A. Rao and M. Georgeff, 'BDI agents from theory to practice', in *Technical note 56*. AAII, (1995).

[27] A. Saptawijaya and L.M. Pereira, 'Towards modeling morality computationally with logic programming', in *16th ISPADL*, pp. 104–119, (2014).

[28] S. Shapiro, Y. Lespérance, and H. J. Levesque, 'The Cognitive Agents Specification Language and Verification Environment for Multiagent Systems', in *2nd AAMAS*, pp. 19–26, (2002).

[29] M. Timmons, *Moral theory: An introduction*, Rowman and Littlefiled, 2012.

[30] M. Tufis and J.-G. Ganascia, 'Normative rational agents: A BDI approach', in *1st Workshop on Rights and Duties of Autonomous Agents*, pp. 37–43. CEUR Proceedings Vol. 885, (2012).

[31] A.F.T. Winfield, C. Blum, and W. Liu, 'Towards and Ethical Robot: Internal Models, Consequences and Ethical Action Selection', in *LNCS*, volume 8717, pp. 85–96, (2014).

[32] M. Wooldridge, N. R. Jennings, and D. Kinny, 'The gaia methodology for agent-oriented analysis and design', *JAAMAS*, **3(3)**, 285–312, (2000).