Consistency and completeness of regulations

Laurence Cholvy[1] and Stéphanie Roussel[1,2]

[1] ONERA Centre de Toulouse
2 avenue Edouard Belin
31055 Toulouse, France
[2] ISAE
10 avenue Edouard Belin,
31055 Toulouse, France

Abstract. In this paper, we deal with regulations that may exist in multi-agent systems in order to regulate agent behaviour. More precisely, we discuss two properties of a regulation, that are its consistency and its completeness. After having defined what consistency and completeness mean, we propose a way to consistently complete incomplete regulations.

Keywords: multi-agent systems, regulations, consistency, completeness

1 Introduction

In a society of agents, a regulation, also called policy, is a set of statements (or norms) which regulate the behaviour of agents by expressing what is obligatory, permitted, forbidden and under which conditions.

Such a regulation is for instance, the one which applies in most countries in EU and which states that *smoking is forbidden in any public area except specific places and in such specific places, smoking is permitted*. Another example of regulation is the one which gives the permissions, prohibitions (and sometimes the obligations) of the different users of a computer system as for file reading, file writing and file execution.

As said before, regulations are means to regulate agent behaviour so that their living together is possible. But, in order to be useful, regulations must be *consistent*, and, in most of the cases, they must also be *complete*.

Consistency is a property of regulations that has already been given some attention in the litterature. For instance, as for confidentiality policies, consistency allows to avoid cases when the user has both the permission and the prohibition to know something ([1]). More generally, according to [3] which studies consistency of general kind of regulations, a regulation is consistent if there exists no possible situation which leads an agent to *normative contradictions* or *dilemmas* also called in [21] *contradictory conflicts* (a given behaviour is prescribed and not prescribed, or prohibited and not prohibited) and *contrary conflicts* (a given behaviour is prescribed and prohibited). Following this definition, consistency of security policies has then been be studied in [4].

At the opposite, completeness of regulations has received much less attention. [1] proposes a definition of completeness between two confidentiality policies (for each piece of information, the user must have either the permission to know it or the prohibition to know it), definition which has been adapted in [5] for multilevel security policies.

Recently, we have studied the notion of completeness for particular regulations which are policies regulating information exchanges in a multi-agent system ([6]). A definition of incompleteness for such policies has been given and a way to reason with incomplete policies has been defined. The approach taken in this work has been rather promising and convinced us to extend this work to any kind of regulations. This is the object of the present paper.

This paper is organised as follows.

Section 2 presents the logical formalism used to express regulations, the definition of consistency of regulations as well as the definition we give of completeness. Section 3 focuses on the problem of reasoning with an incomplete regulation. Following the approach that has led to the CWA (Closed World Assumption) in Database area [16], we present some rules of completion that can be used in order to complete an incomplete regulation. In section 4, we present three examples of regulations. Finally, section 5 is devoted to a discussion and extensions of this work will be mentioned.

2 Regulations

Before studying the properties of regulations, we need to model them. Our proposition is to use a logical formalism so that these properties will be characterized by using the main notions of this formalism.

Modelling general regulations in logic requires at least modelling deontic notions (permission, prohibition, obligation) and modelling individuals and properties on individuals.[1] Then, the ideal logical formalism should at least be a first order modal deontic logic. The complexity of such formalism leads us to restrict the generality of our model. This is why, like many other people before us ([20], [12], [13], [17], [11]...) we choose first order logic as a compromise. More precisely the formalism we consider is a typed first order logic with equality, in which some predicates will be used to model the deontic notions according to the axioms of the simpliest deontic logic, i.e SDL ([2]). This formalism is described below.

2.1 The language

The alphabet of L will be based on four distinct groups of symbols: constant symbols, variable symbols, predicate symbols and function symbols. As we want to type the language, we will distinguish different groups of symbols among those four categories.

Definition 1. *We distinguish two sets of constants: ag-constants (constants for agents), o-constants (other constants) and we distinguish two sets of variables: ag-variables (variables for agents), o-variables (other variables).*

[1] It also should allow to model several dimensions of time (time of validity of norms, deadlines...) and different types of norms (defeasible norms, Contrary-to-duties ...)...

Definition 2. *Predicate symbols are:*

- *d-predicates*[2]*: unary predicates O, P and F (meaning respectively Obligatory, Permitted, Forbidden).*
- *p-predicates: predicates used to express any kind of property.*

Definition 3. *Functions symbols are:*

- *a-functions: used to represent actions of agents. These functions have, at least one argument (say the first one) which represents the agent who performs the action.*
- *not(.): unary-function used to represent object level negation.*

Definition 4. *Terms are defined the following way :*

- *ag-term : ag-constant or ag-variable*
- *o-term : o-constant or o-variable*
- *d-term*[3] *: If f is an a-function with n arguments and if x_1 is a ag-term and $x_2 ... x_n$ are ag-terms or o-terms then $f(x_1, ..., x_n)$ is a d-term. Moreover, if d is a d-term then $not(d)$ is a d-term too.*

Definition 5. *Formulas of L are defined recursively as follows:*

- *Let d be a d-term. Then $O(d)$, $P(d)$, $F(d)$, $\neg O(d)$, $\neg P(d)$ and $\neg F(d)$ are d-literals*[4] *and formulas of L.*
- *If $t_1, ... t_n$ are terms (other than d-terms) and P a p-predicate then $P(t_1, ..., t_n)$ and $\neg P(t_1, ..., t_n)$ are p-literals and formulas of L.*
- *Let F_1 and F_2 be formulas of L and x be a variable. Then $\neg F_1$, $F_1 \wedge F_2$, $F_1 \vee F_2$, $\forall x\ F_1$, $\exists x\ F_1$, $F_1 \rightarrow F_2$ and $F_1 \leftrightarrow F_2$ are formulas of L.*

2.2 Regulation modelling

Definition 6. *A rule is a formula of L which is conjunction of clauses $l_1 \vee l_2 \vee ... \vee l_n$ such that:*

- *(a) l_n is a positive d-literal,*
- *(b) $\forall i \in \{1, ..., n-1\}$, l_i is a p-literal,*
- *(c) if x is a variable in l_n, then $\exists i \in \{1, ..., n-1\}$ such that l_i is a negative literal and contains the variable x.*

In this definition, constraints (a) and (b) allow rules to be conditionals of the form "if such a condition is true then something is obligatory (resp, permitted, forbidden)". Constraint (c) restricts rules to Range Restricted Formulas (We recall that Range-Restricted Formulas are a decidable subset of Domain-Independent Formulas which have been proved to be the only first order formulas having a meaning in modelling [7]).

Definition 7. *A regulation is a set of rules.*

[2] d-predicate stands for "deontic" predicate.
[3] d-term stands for "deontic" term.
[4] d-literal stands for "deontic" literal.

Let us consider an example which will help us to illustrate our purpose all allong section 2 and 3.

Example 1. We consider a regulation which regulates the behaviour of a driver in front of a traffic-light. The language needed is defined as follows :

- A is an ag-constant.
- T is an o-constant.
- x is an ag-variable, t is a o-variable and some other variable will be introduced when needed
- $Green$, $Orange$ and Red are o-constants.
- Car, $Truck$ and $Bike$ are o-constants.
- $D(.)$ is a p-predicate indicating that an agent is a driver.
- $TL(.)$ is a p-predicate indicating that an object is a traffic-light.
- $C(.,.)$ is a p-predicate that takes for parameters a traffic light and a color and indicates the traffic-light color.
- $V(.,.)$ is a p-predicate that takes for parameters a driver and the type of vehicle he drives.
- $IFO(.,.)$ is a p-predicate that takes for parameters an agent and a traffic light and indicates that a driver is in front of a traffic light.
- $S(.,.)$ is an a-function that takes a driver agent and a traffic light for parameters and that indicates whether this agent stops or not in front of the traffic light.

Let's now take the three rules (r_0): *"When a car-driver is in front of a traffic light that is red, he has to stop"*. (r_1): *"When a car-driver is in front of a traffic light that is orange, it is permitted for him to stop"*. (r_2): *"When a car-driver is in front of a traffic light that is green, he must not stop"*. These rules can be modelled by :
$(r_0) : \forall x, t \quad TL(t) \wedge IFO(x,t) \wedge V(x, Car) \wedge C(t, Red) \rightarrow O(S(x,t))$.
$(r_1) : \forall x, t \quad TL(t) \wedge IFO(x,t) \wedge V(x, Car) \wedge C(t, Orange) \rightarrow P(S(x,t))$.
$(r_2) : \forall x, t \quad TL(t) \wedge IFO(x,t) \wedge V(x, Car) \wedge C(t, Green) \rightarrow F(S(x,t))$.

2.3 Back to deontic notions

Since we use a first order logic we must express proper axioms that allow us to reason with the deontic notions. These axioms are the following.

$$\begin{array}{rll} (D) & \forall x & O(not(x)) \rightarrow \neg O(x) \\ (Ax1) & \forall x & F(x) \leftrightarrow O(not(x)) \\ (Ax2) & \forall x & P(x) \leftrightarrow \neg O(not(x)) \wedge \neg O(x) \\ (NO) & \forall x & O(not^{2n}(x)) \leftrightarrow O(x) \end{array}$$

(D) corresponds to the D axiom of SDL. It says that if not performing an action is obligatory then performing this action is not obligatory.

$(Ax1)$ defines the prohibition from the obligation by stating that it is forbidden to perform an action iff it is obligatory not to perform it.

$(Ax2)$ defines the permission by stating that performing an action is permitted iff not performing this action is not obligatory and performing this action is not obligatory neither.

It must be noticed that our definition of permission does not correspond to the usual definition of permission defined in SDL. According to SDL, something is permitted if its negation is not obligatory. However, it has been shown by lawyers [10] that the cases where permission is bilateral (permission to do and permission not to do) are the only valid ones. If not bilateral, permission to do comes to obligation to do[5]. Our definition of bilateral permission corresponds to the notion of optionality[19] (something is optional iff neither it or its negation is obligatory).

Finally, (NO) is introduced because we represent deontic notions in a first order logic and we represent level object negation by the function not.

Definition 8. *We define* $\mathcal{A} = \{(D), (Ax1), (Ax2), (NO)\}$

NOTATION Let $A_1, A_2,$ and A_3 be formulas of L. We note: $A_1 \otimes A_2$ instead of $(A_1 \vee A_2) \wedge \neg(A_1 \wedge A_2)$ and
$A_1 \otimes A_2 \otimes A_3$ instead of $(A_1 \vee A_2 \vee A_3) \wedge \neg(A_1 \wedge A_2) \wedge \neg(A_2 \wedge A_3) \wedge \neg(A_1 \wedge A_3)$. This notation means that one and only one of the formulas A_i is true.

Theorem 1. *For all d-term d, we have*

$$\mathcal{A} \vdash O(d) \otimes P(d) \otimes F(d)$$

2.4 Consistency of regulations

Definition 9. *A state of the world or a world, W, is modelled by a complete set of p-literals (i.e for each p-literal l, we have $l \in W$ or $\neg l \in W$).*

Definition 10. *The set of domain constraints (i.e, constraints which are supposed to be true in any state of the world) is denoted Dom. Dom is a set of formulas of L, written without d-predicates.*

Definition 11. *A state of world W satisfies the constraints Dom iff $W \wedge Dom$ is consistent.*

Definition 12. *Let $\mathcal{R}$ be a regulation, Dom the set of domain constraints and W a world which satisfies Dom. $\mathcal{R}$ is consistent (according to Dom) in W if and only if $W \wedge \mathcal{R} \wedge \mathcal{A}$ is consistent.*

Example 2. $W_0 = \{D(A), TL(T), IFO(A,T), V(A, Car), C(T, Red)\}$.[6]
Dom contains two constraints : (1) a traffic light has a unique color and this color can

[5] For instance, when smoking is permitted, this implies that not smoking is also permitted. If not, that would mean that smoking would be obligatory

[6] For readibility, only positive p-literals are explicitely written.

be green, orange or red, (2) a driver drives one and only one type of vehicle.
$Dom = \{\forall t\ TL(t) \rightarrow C(t, Green) \otimes C(t, Orange) \otimes C(t, Red), \forall x, y, z\ V(x, y) \wedge V(x, z) \rightarrow (y = z)\}$.
W_0 is a world that satisfies Dom.
Let's consider a regulation $\mathcal{R}_0$ that contains the three rules (r_0), (r_1) and (r_2). $W_0 \cup \{O(not^{2n} S(A, T))\}_{n \in \mathbb{N}}$ is a model for $W_0 \wedge \mathcal{R}_0 \wedge \mathcal{A}$. Therefore $W_0 \wedge \mathcal{R}_0 \wedge \mathcal{A}$ is consistent. Thus, $\mathcal{R}_0$ is consistent (according to Dom) in W_0.

Definition 13. *Let $\mathcal{R}$ be a regulation. Dom the set of domain constraints. $\mathcal{R}$ is consistent (according to Dom) iff*
$\forall W \quad W \wedge Dom$ *consistent* $\Rightarrow W \wedge \mathcal{R} \wedge \mathcal{A}$ *consistent*

Proposition 1. *$\mathcal{R}$ is consistent according to Dom iff there is no set of formulas f of L without d-literal such that $f \wedge Dom$ is consistent and $\mathcal{R} \wedge \mathcal{A} \wedge f \wedge Dom$ is inconsistent.*

With this proposition, we show the equivalence between consistency defined in this paper and consistency defined in [3]. Thus, the algorithm proposed there can still be used.

2.5 Completeness of regulations

Informally, a regulation is totally complete as soon as it prescribes the behaviour any agent should have in any situation. We can wonder if this definition really makes sense (can a regulation take into account all possible situations ?). Thus, we suggest to define a partial completeness: a completeness restricted to a formula ϕ and a term ψ: ϕ represents a proposition and ψ an action regulated. Thus, we want a regulation be complete for ϕ and ψ iff in any situation where ϕ is true, it is obligatory (resp. permitted, forbidden) for an agent to perform ψ.

In what follows, we will use X to express a n-tuple of $ag - terms$ and $o - terms$.

This leads to the following definition:

Definition 14. *Let $\mathcal{R}$ be a regulation, W a world regulated by $\mathcal{R}$ and consistent with Dom, $\phi(X)$ a formula of L without d-literal and $\psi(X)$ a d-term. $\mathcal{R}$ is complete in W for $\vdash$, $\phi(X)$ and $\psi(X)$ if and only if, for all X*

$$\textit{If } W \vdash \phi(X) \textit{ Then}$$

$$\begin{aligned}(W \wedge \mathcal{R} \wedge \mathcal{A} &\vdash O(\psi(X)) \textit{ or}\\ W \wedge \mathcal{R} \wedge \mathcal{A} &\vdash P(\psi(X)) \textit{ or}\\ W \wedge \mathcal{R} \wedge \mathcal{A} &\vdash F(\psi(X)))\end{aligned}$$

Example 3. Let's consider the world
$W_0 = \{D(A), TL(T), IFO(A, T), V(A, Car), C(T, Red)\}$. Consider the regulation $\mathcal{R}_0$.
Let's take $\phi_0(x, t) = IFO(x, t)$ and $\psi_0(x, t) = S(x, t)$.

$W_0 \vdash IFO(A,T)$ and $W_0 \wedge \mathcal{R}_0 \wedge \mathcal{A} \vdash O(S(A,T))$. Thus, $W_0 \vdash \phi_0(A,T)$ and $W_0 \wedge \mathcal{R}_0 \wedge \mathcal{A} \vdash O(\psi_0(A,T))$ and $\mathcal{R}_0$ is complete in W_0 for $\vdash$, $\phi_0(x,t)$ and $\psi_0(x,t)$.

Let's now consider the world
$W_1 = \{D(A), TL(T), IFO(A,T), V(A, Truck), C(T, Red)\}$. W_1 is consistent with Dom.
$W_1 \vdash IFO(A,T)$ but $W_1 \wedge \mathcal{R}_0 \wedge \mathcal{A} \nvdash O(\psi_0(A,T))$ and $W_1 \wedge \mathcal{R}_0 \wedge \mathcal{A} \nvdash P(\psi_0(A,T))$ and $W_1 \wedge \mathcal{R}_0 \wedge \mathcal{A} \nvdash F(\psi_0(A,T))$.
Thus, $\mathcal{R}_0$ is incomplete in W_1 for $\vdash$, $\phi_0(x,t)$ and $\psi_0(x,t)$. In fact, no rule of the regulation can be applied as the vehicle isn't a car but a truck.

This definition can be generalized as follows :

Definition 15. *Let $\mathcal{R}$ be a regulation. $\mathcal{R}$ is complete for $\vdash$, $\phi(X)$ and $\psi(X)$ if and only if for all world W consistent with Dom, $\mathcal{R}$ is complete for $\vdash$, $\phi(X)$ and $\psi(X)$ in W.*

Completeness is an important issue for a regulation. For a given situation, without any behaviour stipulated, any behaviour could be observed and thus consequences could be quite important. With an incomplete regulation, we could (1) detect the "holes" of the regulation and send them back to the regulation designers so that they can correct them or (2) detect the "holes" of the regulation and allow for those holes some completion rules that could be applied to correct them. The first solution could be quite irksome to be applied (the number of holes could be quite important and thus correct them one by one quite long). Then, we put in place the second solution.

3 Reasoning with incomplete regulations

3.1 Completion rules

In this paragraph, we present a solution which extends the CWA defined by Reiter to complete first order databases.

According to CWA, if the database is incomplete for a literal l (i.e l is not deduced in the database), then it can be assumed that its negation ($\neg l$) is deduced. This rule is motivated by the assumption that a database is used to represent the real world. Since in the real world, a fact is true or is false (i.e $l \otimes \neg l$ is a tautology in first order logic) then a database must deduce a fact or its negation.

Here, given a d-term l, we are not interested in its truth value but in the fact that a given regulation deduces that it is obligatory, forbidden or tolerated. These three cases are the only ones because axioms $\mathcal{A}$ imply $O(l) \otimes F(l) \otimes P(l)$. Thus, if the regulation is incomplete for a literal l (i.e it does not deduce neither $O(l)$ nor $F(l)$ nor $P(l)$) then it can only be completed by assuming that $O(l)$ can be deduced, or $P(l)$ or $F(l)$. This leads to the three completion rules which are described in the following.

Furthermore, in order to be as general as possible, we define parameterised completion rules so that the way of completing by $O(l)$, $P(l)$ or $F(l)$ may depend on some conditions on the current world.

Let $\mathcal{R}$ be a consistent regulation and W be a world regulated by $\mathcal{R}$.

Notation. For more readibility, we will write *"$\mathcal{R}, W$ incomplete for X"* instead of: $W \vdash \phi(X)$ and $\mathcal{R}, W, \mathcal{A} \nvdash O(\psi(X))$ and $\mathcal{R}, W, \mathcal{A} \nvdash P(\psi(X))$ and $\mathcal{R}, W, \mathcal{A} \nvdash F(\psi(X))$.

Let E_F, E_P and E_O be three formulas that depend on X.

The three inference rules[7] are:

$$(R_{E_F}) \qquad \frac{\mathcal{R}, W \ incomplete\ for\ X, \quad W \vdash E_F(X)}{F(\psi(X))}$$

$$(R_{E_P}) \qquad \frac{\mathcal{R}, W \ incomplete\ for\ X, \quad W \vdash E_P(X)}{P(\psi(X))}$$

$$(R_{E_O}) \qquad \frac{\mathcal{R}, W \ incomplete\ for\ X, \quad W \vdash E_O(X)}{O(\psi(X))}$$

We can complete an incomplete regulation so that $\psi(X)$ is forbidden (R_{E_F}), permitted (R_{E_P}) or obligatory (R_{E_O}) depending on $E_F(X)$, $E_P(X)$ and $E_O(X)$. We define here a **new inference** that we will note $\vdash_*$[8] . Rules of inference for $\vdash_*$ are the same as for $\vdash$ but we add R_{E_F}, R_{E_P} and R_{E_O}.

The next step is to define the conditions for which the regulation is complete and consistent with this new inference. This will be addressed in the next section.

3.2 Consistency and completeness of the completed regulation

Extending definitions First of all, Then, we have to extend the definition of consistency for the new inference.

Definition 16. *Let W be a world consistent with Dom and $\mathcal{R}$ a regulation that is consistent for $\vdash$ in W* [9]*. $\mathcal{R}$ is consistent for $\vdash_*$ in W (according to domain Dom) if and only if $W \wedge \mathcal{R} \wedge \mathcal{A}$ is consistent for $\vdash_*$.*

Then we have to extend the definition of completeness of a regulation with the inference $\vdash_*$.

Definition 17. *Let $\mathcal{R}$ be a regulation and W a world regulated by $\mathcal{R}$ and consistent with Dom . $\mathcal{R}$ is complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$ in W if and only if we have : for all X*

$$\textit{If } W \vdash \phi(X) \textit{Then}$$

$$\begin{gathered}(\mathcal{R}, W, \mathcal{A} \vdash_* O(\psi(X)) \textit{ or}\\ \mathcal{R}, W, \mathcal{A} \vdash_* P(\psi(X)) \textit{ or}\\ \mathcal{R}, W, \mathcal{A} \vdash_* F(\psi(X)))\end{gathered}$$

This definition can be generalized.

[7] These rules should be indexed by ϕ and ψ but for readibility reasons, this is omitted

[8] This inference should be indexed by ϕ and ψ but for readibility reasons, this is omitted

[9] It's not relevant to study a regulation that is not consistent in W

Definition 18. *Let $\mathcal{R}$ be a regulation. $\mathcal{R}$ is complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$ if and only if for all world W consistent with Dom, $\mathcal{R}$ is complete in W for $\vdash_*$, $\phi(X)$ and $\psi(X)$.*

Sufficient and necessary condition

Proposition 2. *Let $\mathcal{R}$ be a regulation and W a world regulated by $\mathcal{R}$ and consistent with Dom. $\mathcal{R}$ is complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$ in W if and only if for all X*

$$\mathcal{R}, W \text{ incomplete for } X \Rightarrow W \vdash E_F(X) \vee E_P(X) \vee E_O(X)$$

Proposition 3. *A regulation $\mathcal{R}$ that is complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$ in a world W consistent with Dom [10] is consistent for $\vdash_*$ in W (according to Dom) if and only*

$$\forall X \quad \text{If } \mathcal{R}, W \text{ incomplete for } X \text{ Then}$$

$$W \vdash \neg(E_F(X) \wedge E_P(X)) \wedge \neg(E_F(X) \wedge E_O(X)) \wedge \neg(E_P(X) \wedge E_O(X))$$

Corollary 1. *Let $\mathcal{R}$ be a regulation and W a world consistent regulated by $\mathcal{R}$ with Dom. $\mathcal{R}$ is consistent and complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$ in W if and only if for all X*

$$\text{If } \mathcal{R}, W \text{ incomplete for } X \text{ Then } W \vdash E_F(X) \otimes E_P(X) \otimes E_O(X)$$

Example 4. Let's consider the world
$W_1 = \{D(A), TL(T), IFO(A,T), V(A, Truck), C(T, Red)\}$ from the last example.
$\mathcal{R}_0, W_1$ is incomplete for (A, T).
Let's take $E_F(x,t) = V(x, Truck) \wedge C(t, Green)$, $E_P(x,t) = V(x, Truck) \wedge C(t, Orange)$ and
$E_O(x,t) = V(x, Truck) \wedge C(t, Red)$.
$W_1 \vdash E_O(A, T)$. Thus, $\mathcal{R}_0$ is consistent and complete for $\vdash_*$, $\phi_0(x,t)$ and $\psi_0(x,t)$ in W_1.

This corollary characterizes necessary and sufficient conditions for the three completion rules (R_{E_O}), (R_{E_P}) and (R_{E_F}) to consistently complete an incomplete regulation. More precisely, this corollary says that if every time the regulation does not prescribe a behaviour one and only one E_i is true, then the three completion rules consistently complete the regulation (because one and only one completion rule is applied).

Yet, even if this necessary and sufficent condition is interesting in theory, it is not really useful for practical purposes. In fact, to verify that this condition is satisfied, we would have to detect every "hole" in the regulation. This detection is an operation we want to avoid. Thus, we try to find more general conditions that are still sufficient but not necessary for the completion rules to consistently complete the regulation.

The following section addresses this point and defines some sufficient but not necessary conditions.

[10] More precisely, we should say that a regulation $\mathcal{R}$ is complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$ in a world W consistent with Dom and for the $E_F(X)$, $E_O(X)$, $E_P(X)$ formulas.

Sufficient conditions

Proposition 4. *Let $\mathcal{R}$ be a regulation and W a world regulated by $\mathcal{R}$ and consistent with Dom. If*

$$\forall X \; W \vdash \phi(X) \rightarrow E_O(X) \otimes E_F(X) \otimes E_P(X)$$

then $\mathcal{R}$ is consistent and complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$ in W.

Example 5. Let's consider the world $W_2 = \{D(A), TL(T), IFO(A,T), V(A, Bike), C(T, Red)\}$. W_2 is consistent with Dom.
Consider the regulation $\mathcal{R}_0$. This time, let's take $E_F(x,t) = C(t, Green)$, $E_P(x,t) = C(t, Orange)$ and $E_O(x,t) = C(t, Red)$. $W_2 \vdash \phi_0(A,T) \rightarrow E_O(A,T)$. Thus, $\mathcal{R}_0$ is consistent and complete for $\vdash_*$, $\phi_0(X)$ and $\psi_0(X)$ in W_2. But we have $W_1 \vdash \phi_0(A,T) \rightarrow E_O(A,T)$ too so $\mathcal{R}_0$ is consistent and complete for $\vdash_*$, $\phi_0(X)$ and $\psi_0(X)$ in W_1. Those more general E_i allow us to have a regulation complete for any type of vehicle.

We can generalize the last proposition to all worlds consistent with Dom.

Proposition 5. *Let $\mathcal{R}$ be a regulation. If*

$$\forall X \; \vdash Dom \rightarrow (\phi(X) \rightarrow E_O(X) \otimes E_F(X) \otimes E_P(X))$$

then $\mathcal{R}$ is consistent and complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$.

Corollary 2. *Let $\mathcal{R}$ be a regulation. If*

$$\forall X \; \vdash Dom \rightarrow E_O(X) \otimes E_F(X) \otimes E_P(X)$$

then $\mathcal{R}$ is consistent and complete for $\vdash_*$, $\phi(X)$ and $\psi(X)$.

Example 6. $\forall(x,t) \vdash Dom \rightarrow C(t, Red) \otimes C(t, Green) \otimes C(t, Orange)$. Thus, $\mathcal{R}_0$ is consistent and complete for $\vdash_*$, $\phi_0(X)$ and $\psi_0(X)$.
In fact, Dom specifies that a traffic light has one and only one color among three colors Red, $Orange$ and $Green$. If there is one E_i for each color, we are sure that whatever the situation is, we can apply one and only one E_i if there is a "hole" in the regulation.

Another alternative would be to take fixed E_i. For example, we could take one E_i equal to $True$ and the two others to $False$.

- Let's take for example $E_F = True$, $E_P = False$ and $E_O = False$. In this case, according to completion rules, everything that is not specified as obligatory or permitted by the regulation is forbidden. This strict behaviour could be observed for regulations that regulate a highly secured system where each action has to be explicitely authorized before being performed.
- Another case could be $E_F = False$, $E_P = True$ and $E_O = False$. We are here in the opposite situation, meaning that everything that is not obligatory or forbidden is permitted. This "tolerant" behaviour could be observed for regulations for dimmed secured systems where everything that is not forbidden or obligatory is implicitly permitted. This case is strongly related to the "sealing legal principle" that has been studied in [18].
- Finally, the last case is $E_F = False$, $E_P = False$ and $E_O = True$. In this case, every action that is not forbidden or permitted has to be performed.

4 Examples of regulations

4.1 Information exchange policies

Consistency and completeness of information exchange policies An information exchange policy is a regulation which prescribes the behaviour of agents in a multi-agent system regarding information communication.

To describe such policies, we need one particular binary p-predicate $Receive$ and one particular ternary function $tell$.

$Receive(x, i)$ means that agent x receives information i.

$tell(x, i, y)$ represents the event created by an agent x making the action of telling agent y a piece of information i.

The consistency of such policies is defined by definition 12.

The completeness of such policies is defined by instanciated definition 14 of section 2.4 with the following specific formula:

$$\phi(x, i, y) = Receives(x, i) \wedge Agent(y) \wedge \neg(x = y)$$

and the specific term

$$\psi(x, i, y) = tell(x, i, y)$$

This leads to the following definition:

Definition 19. *Let $\mathcal{P}$ be a information exchange policy and W a world regulated by $\mathcal{P}$. $\mathcal{P}$ is complete for $\vdash$ in W iff, for all $X = (x, y, i)$*

$$\textit{If } W \vdash Receive(x, i) \wedge Agent(y) \wedge \neg(x = y) \textit{ Then}$$

$$\mathcal{P}, W, \mathcal{A} \vdash O(tell(X)) \quad \textit{or}$$

$$\mathcal{P}, W, \mathcal{A} \vdash F(tell(X)) \quad \textit{or}$$

$$\mathcal{P}, W, \mathcal{A} \vdash P(tell(X))$$

Notation introduced in section 3 is reformulated as follows:

Notation. *"$\mathcal{P}, W$ incomplete for (x, i, y)"* is for:
$W \vdash Receive(x, i) \wedge Agent(y) \wedge \neg(x = y)$ and $\mathcal{P}, W, \mathcal{A} \nvdash O(tell(x, i, y))$ and $\mathcal{P}, W, \mathcal{A} \nvdash P(tell(x, i, y))$ and $\mathcal{P}, W, \mathcal{A} \nvdash F(tell(x, i, y))$.

Thus, completion rules are the following:

$$(R_{E_F}) \qquad \frac{\mathcal{P}, W\ incomplete\ for\ X, \quad W \vdash E_1(X)}{F(tell(X))}$$

$$(R_{E_P}) \qquad \frac{\mathcal{P}, W\ incomplete\ for\ X, \quad W \vdash E_2(X)}{P(tell(X))}$$

$$(R_{E_O}) \qquad \frac{\mathcal{P}, W\ incomplete\ for\ X, \quad W \vdash E_3(X)}{O(tell(X))}$$

Results proved in section 3 remain valid. In particular, we still have the three cases:

- $E_F = True$, $E_P = False$ and $E_O = False$.
  This applies to highly secured multi-agent systems in which any communication action should be explicitely obligatory or permitted before being performed.
- $E_F = False$, $E_P = True$ and $E_O = False$.
  This case applies to lowly secured system in which any communication action which is not explicitely forbidden is permitted.
- $E_F = False$, $E_P = False$ and $E_O = True$.
  In this case, unless explicit mentioned, sending information is obligatory.

Illustration In order to illustrate this, consider the example of an enterprise in which there is a manager and two employees. Consider a policy $\mathcal{P}_0$with only one rule which states that *"Managers are required not to inform their employees about any equipment checking information"*.

$$(R_0) \quad \forall(x, y, i) \quad Manager(x) \wedge Employee(y)$$

$$\wedge Receive(x, i) \wedge Topic(i, EqtChk) \rightarrow F(tell(x, i, y))$$

Take $Dom = \{\}$ and consider the following world:

$$W_0 = \{Agent(a), Agent(b), Manager(a), Employee(b)$$

$$Topic(i_1, ExpRisk), Receive(a, i_1)\}.$$

In this situation, a is a manager, b an employee. a has received information i_1 whose topic is Explosion Risk.

$(W_0, Dom, \mathcal{P}_0, \mathcal{A})$ is consistent. Thus, $\mathcal{P}_0$ is consistent in W_0.

However we have $W_0 \vdash Receive(a, i_1) \wedge Agent(b) \wedge \neg(a = b)$ but $\mathcal{P}_0, W_0, \mathcal{A} \nvdash O(tell(a, i_1, b))$ and $\mathcal{P}_0, W_0, \mathcal{A} \nvdash T(tell(a, i_1, b))$ and $\mathcal{P}_0, W_0, \mathcal{A} \nvdash F(tell(a, i_1, b))$. Thus, $\mathcal{P}_0$ is incomplete for $\vdash$.

Here, incompleteness comes from the fact that the policy prescribes the behaviour of the boss if he/she receives an information about Equipment Verification but it does not prescribe anything as for information about Explosion Risk. To say it differently, the policy does not state what the boss should do when he receives information about Risk Explosion.

In order to complete the previous policy, we could take:

$E_F(x, y, i) = Topic(i, EqtChk)$, $E_P(x, y, i) = False$ and $E_O(x, y, i) = Topic(i, ExpRisk)$. This comes to force the boss to tell its employees about Risk Exploxion information. We can verify that $\mathcal{P}_0$ is complete and consistent for $\vdash_*$ in W_0.

Let consider now that Dom contains a constraint *"An information has one and only one topic and this topic can be EqtChk, ExpRisk, Meeting or EqtOutOfOrder"*. Take:

$E_F(x, y, i) = Topic(i, EqtChk) \vee Topic(i, Meeting)$,
$E_P(x, y, i) = Topic(i, EqtOutOfOrder)$ and
$E_O(x, y, i) = Topic(i, ExpRisk)$

We can apply the corollary 2 to conclude that $\mathcal{P}_0$ is complete and consistent for $\vdash_*$.

4.2 Security policies

Security policies are regulations which regulate computer systems access. More precisely, they prescribe the behaviour of the system users concerning the files reading, or writing or execution.

The definition of completeness given in section 2 is very general and it can be instanciated in several ways to be applied to security policies.

For instance, we can consider:

$$\phi_1(x,y) = User(x) \wedge Permanent(x) \wedge File(y)$$

$$\phi_2(x,y) = User(x) \wedge Temporary(x) \wedge File(y)$$

$$\psi_1(x,y) = read(x,y)$$

$$\psi_2(x,y) = write(x,y)$$

For instance, a security policy may be complete (in a world W) for $\phi_1(x,y)$ and $\psi_1(x,y)$, $\phi_1(x,y)$ and $\psi_2(x,y)$ but may be incomplete for $\phi_2(x,y)$ and $\psi_1(x,y)$, $\phi_2(x,y)$ and $\psi_2(x,y)$.

This means that the policy completely prescribes the behaviour of permanent users regarding reading and writing files, but is incomplete as for temporary users and reading or writing files.

5 Conclusion

In this paper, we addressed the problem of analysing consistency and completeness of regulations which may exist in a society of agents in order to regulate their behaviour.

More specifically, we have defined a logical framework and showed how to express a regulation within this framework. Then, consistency and completeness for a regulation have been defined. The definition of completeness we gave is rather general and we showed how to instanciate it on two particular examples. This constitutes the first contribution of our work.

We also dealt with incomplete regulations and proposed a way for completing them by using three inference rules. We have established several results which showed when these rules consistently complete a regulation. This constitutes the second contribution of our work. Let us add that this proposal, even we did not develop it here, can be reformulated within the framework of Reiter's defaults [14].

The notion of completeness developed here is in fact a kind of "local completeness", in the sense that for proposition $\phi(X)$ only, we require to have $O(\psi(X))$, $P(\psi(X))$ or $F(\psi(X))$. That looks close to the notion of completeness introduced in the Database domain by [15] and [8], who noticed that some of the integrity constraints that are expressed on a database are what the database should know (Or, to say it differently, these are rules about what should be deduced in the database). For instance, the integrity constraint expressing that "any employee has got a phone number, a fax number or a mail adress" expresses in fact that, for any employee known by the database, the database knows its phone number, its fax number or its mail address[11]. As first mentioned by Reiter [15], this integrity constraint expresses a kind of local completeness of the database. Reiter's defaults can be used in order to complete such a database in case of incompleteness. For instance, one of the rules can be that if the database does contain any required information (no phone number, no fax number, no mail address) for a given employee but if the department that employee works in is known, then it can be assumed that its phone number is the phone number of its department.

Studying the formal link between the notion of completeness introduced in this paper and that notion of local completeness constitutes one interesting extension of this work.

Furthermore, in order to deal with more general regulations, this present work must be extended. In particular, we have to extend it by considering the notion of time. Indeed, as it is shown in [9], this issue is very important when speaking about obligations and we will have to consider different types of time among which, at least, the time of validity of the norms and the deadlines beared on the obligations.

Let us say that we forsee to reformulate the work presented here by using a deontic modal logic. This will allow us to deal with more general kind of regulations by allowing us to imbricate modalities of, for instance obligation and time.

References

1. P. Bieber and F. Cuppens. Expression of confidentiality policies with deontic logic. In *Proceedings of the First Workshop on Deontic Logic and Computer Science (DEON'91)*, 1991.
2. B. F. Chellas. *Modal logic, an introduction*. Cambridge University Press, 1980.
3. L. Cholvy. Checking regulation consistency by using SOL-resolution. In *International Conference on Artificial Intelligence and Law*, pages 73–79, 1999.
4. L. Cholvy and F. Cuppens. Analysing consistency of security policies. In *IEEE Symposium on Security and Privacy*, 1997.
5. F. Cuppens and R. Demolombe. A modal logical framework for security policies. In *Lectures Notes in Artificial Intelligence, 1325*. Springer, 1997.
6. L. Cholvy, S. Roussel Reasoning with incomplete information exchange policies In *Proc of ninth European Conference of Symbolic and Quantitative Approaches to Uncertainty, ECSQARU'07, Hammamet, october 2007.*
7. R. Demolombe. Syntactical Characterization of a subset of Domain Independent Formulas. In *Journal of ACM, 39(1)*, 1982.
8. R. Demolombe. Database validity and completeness: another approach and its formalisation in modal logic. In *Proc. IJCAI Workshop on Knowledge representation meets databases*, 1999.
9. R. Demolombe, P. Bretier, and V. Louis. Norms with deadlines in Dynamic Deontic Logic In *Proc. ECAI 2006*, 751-752, 2006.
10. C. Groulier. Normes permissives et droit public Thèse de l'Université de Limoges, 2006. http://www.unilim.fr/scd/theses/accesdoc.html

[11] Notice that this does not prevent the fact that in the real world, an employee of the company has no telephone number, no fax number and no mail address.

11. J.Y. Halpern, V. Weissman Using First-Order Logic to Reason about Policies Proc. of the 6^{th} IEEE Computer Security Foundations Workshop CSFW-03, 2003.
12. R. Lee Bureaucracies as deontic systems *ACM Transactions on Information Systems (TOIS)* 6(2), pp 87 - 108 , April 1988.
13. Ong, R. Lee Detecting deontic dilemmas in bureaucratic rules: a first-order implementation using abduction Proc of *Proceedings of the second Workshop on Deontic Logic and Computer Science (DEON'94)*, Oslo, 1994.
14. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2), 1980.
15. R. Reiter. What should a database know? *Journal of Logic Programming*, 14(1-2):127–153, 1992.
16. R. Reiter. On closed world data bases. In J.M. Nicolas H. Gallaire, J. Minker, editors, *Logic and Databases*. Plenum Publications, New-York, 1978.
17. U. Ryu, R.M. Lee. Defeasible Deontic Reasoning and Its Applications to Normative Systems. *Decision Support Systems*, 14(1), pp. 59-73, 1995.
18. L. M. M. Royakkers. Giving permission implies giving choice *Proceedings of the Eighth International Workshop on Database and Expert Systems Applications*, pp. 198-203, 1997.
19. Stanford Encyclopedia of Philosophy. http://plato.stanford.edu/entries/logic-deontic/
20. M.J. Sergot. Prospects for representing the law as logic programs. . In *K.L. Clark and S.Â. Tarnlund, editors, Logic Programming* , pp 33–42. .Academic Press, London, 1982.
21. E. Vranes. The definition of ”norm conflict” in international law and legal theory. *The European Journal of International Law*, 17(2):395–418, 2006.