To cite this document: Cholvy, Laurence and Garion, Christophe and Roussel, Stéphanie *Cohérence et complétude des réglementations en présence de contraintes.* (2012) In: Sixièmes Journées d'Intelligence Artificielle Fondamentale (JIAF), 22-24 may 2012, Toulouse, France.

Cohérence et complétude des réglementations en présence de contraintes

Laurence Cholvy[1] **Christophe Garion**[2] **Stéphanie Roussel**[3]
[1] ONERA - Toulouse
[2] Université de Toulouse - ISAE/DMIA
[3] CRIL - CNRS/Université d'Artois - Lens
cholvy@onera.com garion@isae.fr sroussel@cril.fr

Abstract

Cet article a pour objet l'étude des réglementations qui existent dans les systèmes multi-agents et qui visent à réguler le comportement des agents. Plus précisément, nous nous intéressons à deux propriétés des réglementations : la cohérence et la complétude, et ce en présence de contraintes (lois de la nature, faits persistants) dont nous montrons l'influence. Nous montrons également comment compléter une réglementation incomplète toute en préservant sa cohérence. Ce travail considère que les réglementations sont exprimées en logique déontique du premier ordre.

1 Introduction

Les réglementations sont des composants des systèmes multi-agents dont le but est de réguler le comportement des agents. Ce sont des ensembles d'énoncés qui expriment ce qui est obligatoire, permis, interdit et sous quelles conditions. Par exemple, la réglementation suivante s'applique dans la plupart des pays européens : « il est interdit de fumer dans les lieux publics, sauf sans certains endroits spécifiques où cela est permis ». Mais, pour être utile, une réglementation doit être *cohérente* et, la plupart du temps, elle doit aussi être *complète*.

La cohérence des réglementations a fait l'objet de nombreuses études. Par exemple, dans le domaine plus précis des politiques de confidentialité, la cohérence d'une politique de confidentialité vise à éviter des cas où connaître une information est à la fois permis et interdit selon [3]. Plus généralement, selon [5], une réglementation est cohérente s'il n'existe pas de situation possible dans laquelle un agent serait face à (1) *des contradictions normatives*, *des dilemmes* (une attitude est à la fois obligatoire et non obligatoire ou interdite et non interdite) ou (2) *des conflits* [18] (une attitude est à la fois obligatoire et interdite).

La seconde propriété souvent requise pour les réglementations est la complétude. Informellement, une réglementation est complète si elle prescrit toute attitude de tout agent dans toute situation. Cette propriété est un élément central de la théorie des contrats [11] et en droit, où le *sealing legal principle* stipule que « tout ce qui n'est pas interdit est permis » [17] et est une façon de compléter une réglementation. La complétude des réglementations a fait l'objet de peu de travaux en informatique ou plus spécifiquement en intelligence artificielle. [3] propose une définition de la complétude entre deux politiques de confidentialité : pour toute information donnée, connaître cette information est soit autorisé soit interdit. Cette définition a ensuite été adaptée dans [6] pour des politiques multiniveaux.

La plupart des réglementations s'appliquent dans des environnement contraints i.e. des environnements où des informations sont supposées vraies. Comme il est mentionné dans [10], ces contraintes doivent être prises en compte lorsqu'on étudie la cohérence des réglementations. Nous pensons ici que deux types d'informations différentes peuvent être considérées comme contraintes :

– tout d'abord, les lois de la nature ou *contraintes d'intégrité*, parce qu'elles sont nécessairement vraies. Par exemple, « avancer et s'arrêter sont deux actions que l'on ne peut pas réaliser en même temps » est une de ces contraintes.

– les *faits persistants*. Par exemple, « tel agent est un conducteur », « tel objet est une voiture » sont des faits persistants.

Nous montrerons que la cohérence et la complétude des réglementations sont influencées par ces contraintes. Par exemple, sous la contrainte « avancer et s'arrêter sont deux actions que l'on ne peut pas réaliser en même temps », une réglementation qui obligerait un agent à avancer et à s'arrêter en même temps serait incohérente.

[9] a proposé une logique modale du premier ordre pour modéliser les réglementations et étudier leur cohérence et complétude. Ce travail prenait déjà en compte des contraintes d'intégrité mais ces contraintes ne contraignaient pas les actions qui sont réglementées : par exemple, la contrainte mentionnée ci-dessus ne pouvait pas être prise en compte dans ce travail. Cette restriction a par ailleurs conduit à une partition de l'ensemble des prédicats du langage assez compliquée.

Notre objectif est ici d'étendre ce travail en considérant des contraintes générales sans partition dans le langage. Nous étudierons ensuite une inférence particulière pour rendre complètes et cohérentes les réglementations incomplètes et ce en présence de contraintes.

Cet article est organisé comme suit. La section 2 présente le formalisme logique utilisé pour exprimer les réglementations. La modélisation des réglementations et de leurs propriétés est présentée dans la section 3. La section 4 montre comment raisonner avec une réglementation incomplète. Pour finir, la section 5 sera consacrée à une discussion et présentera quelques extensions possible de ce travail.

2 Formalisme utilisé : FOSDL

Le formalisme de base pour modéliser les réglementations est la logique modale propositionnelle SDL (*Standard Deontic Logic*) [4]. Ici, nous étendons SDL en FOSDL (*First-Order Standard Deontic Logic*) afin d'exprimer des réglementations plus complexes impliquant plusieurs agents. Cette extension est faite selon les idées développées dans [8].

2.1 Langage

L'alphabet du langage de FOSDL est basé sur les ensembles suivants de symbols non logiques :

– un ensemble $\mathcal{P} = \{Q, R, ...\}$ de symboles de prédicats

– un ensemble $\mathcal{F} = \{f, g, ...\}$ de symboles de fonctions

– un symbole de modalité O représentant l'obligation

L'ensemble des fonctions d'arité 0 est appelé *ensemble des constantes* et est dénoté par $\mathcal{C}o$.

On définit également dans l'alphabet de FOSDL des symboles logiques :

– un ensemble $\mathcal{V}$ de symboles de variables

– les connecteurs classiques $\neg$, $\vee$ et le quantificateur $\forall$

– les symboles (et)

On appelle *terme* une variable de $\mathcal{V}$ ou l'application d'un symbole de fonction à des termes. Notons que les constantes sont aussi des termes. Dans ce qui suit, on utilisera des lettres latines majuscules pour les symboles de prédicats, des lettres latines minuscules pour les fonctions et $\{x_1, \ldots, x_i, \ldots\}$ pour les variables.

Définition 1 (Formules de FOSDL)
Les formules de FOSDL sont définies récursivement par :

– *si $t_1, \ldots, t_n$ sont des termes et Q un symbole de prédicat n-aire alors $Q(t_1, \ldots, t_n)$ est une formule de FOSDL.*

– *si φ est une formule de FOSDL, alors $O\varphi$ est une formule de FOSDL. $O\varphi$ signifie « φ est obligatoire ».*

– *si ψ_1 et ψ_2 sont des formules de FOSDL et x_1 un symbole de $\mathcal{V}$, alors $\neg\psi_1$, $\psi_1 \vee \psi_2$, $\forall x_1\ \psi_1$ sont des formules de FOSDL.*

On appelle *littéral positif* toute formule de la forme $Q(t_1, \ldots, t_n)$ et *littéral négatif* toute formule de la forme $\neg Q(t_1, \ldots, t_n)$, Q étant un symbole de prédicat et $t_1, \ldots, t_n$ des termes.

Si φ_1, φ_2 et φ_3 sont des formules de FOSDL et x_1 est un symbole de variable, on définit aussi les abréviations suivantes :

$\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2)$
$\varphi_1 \otimes \varphi_2 \otimes \varphi_3 \equiv (\varphi_1 \wedge \neg\varphi_2 \wedge \neg\varphi_3) \vee (\neg\varphi_1 \wedge \varphi_2 \wedge \neg\varphi_3) \vee (\neg\varphi_1 \wedge \neg\varphi_2 \wedge \varphi_3)$
$\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$
$\varphi_1 \leftrightarrow \varphi_2 \equiv (\neg\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg\varphi_2)$
$\exists x_1\ \varphi_1 \equiv \neg\forall x_1\ \neg\varphi_1$

Les modalités de permission, notée P, et d'interdiction, notée F, sont définies à partir de O par $F\varphi \equiv O\neg\varphi$ et $P\varphi \equiv \neg O\varphi \wedge \neg O\neg\varphi$.

On peut remarquer que la définition de la permission que nous avons choisie correspond à la notion de *permission facultative* ou *bilatérale* [1]. Une permission qui n'est pas bilatérale revient en effet à une obligation [1]. Notre définition de permission bilatérale correspond à la notion d'*optionalité* [14] (une action est

1. Par exemple, lorsqu'il est permis de fumer alors il est également permis de ne pas fumer, car sinon, cela voudrait dire qu'il est obligatoire de fumer.

optionnelle si et seulement si elle n'est pas obligatoire et sa négation ne l'est pas non plus). On peut enfin remarquer que les définitions données pour l'obligation, la permission et l'interdiction correspondent aux trois positions normatives définies par Kanger et Lindahl dans [12, 13].

Une formule de FOSDL sans modalité est dite *objective*. Les termes et formules de FOSDL sans symboles de variable sont dits *de base*. L'ensemble des termes de base dans FOSDL est appelé l'univers de Herbrand et est notéHU.

Enfin, on appelle *substitution de base* toute fonction $\chi\ : \mathcal{V} \rightarrow HU$. si $\varphi(x)$ est une formule de FOSDL avec x comme variable libre, $\varphi(\chi(x))$ est la formule φ dans laquelle les occurrences de x ont été remplacées par $\chi(x)$.

2.2 Axiomatique

On définit un système d'inférence pour FOSDL en savant l'approche présentée dans [8]. Dans ce qui suit, $\varphi(x)$ dénote une formule dans laquelle la variable x a des occurrences libres. On dit que la variable libre y est *substituable* à x dans $\varphi(x)$ si aucune occurrence libre de x dans $\varphi(x)$ n'est dans la portée de $\forall y$ dans $\varphi(x)$.

Définition 2 (Axiomes et règles d'inférence)
Les formules suivantes sont des axiomes de FOSDL :

(Taut)	*Toutes les tautologies de FOL*
(UnDis)	$(\forall x\ (\varphi \rightarrow \psi)) \rightarrow (\forall x\ \varphi \rightarrow \forall x\ \psi)$
(Perm)	$\forall x \forall y\ \varphi \leftrightarrow \forall y \forall x\ \varphi$
(UnInst)	$\forall y\ (\forall y\ \varphi(x) \rightarrow \varphi(y))$ *où y est substituable à x dans $\varphi(x)$*
(OK)	$O(\varphi \rightarrow \psi) \rightarrow (O\varphi \rightarrow O\psi)$
(OD)	$O\varphi \rightarrow \neg O \neg \varphi$
(OBar1)	$O(\forall x\ \varphi) \rightarrow \forall x\ O\varphi$
(OBar2)	$\forall x\ O\varphi \rightarrow O(\forall x\ \varphi)$

Les règles d'inference de FOSDL sont :

(MP) $\dfrac{\varphi \quad \varphi \rightarrow \psi}{\psi}$ *(Gen)* $\dfrac{\varphi}{\forall x\ \varphi}$

(ONec) $\dfrac{\varphi}{O\varphi}$

Une sémantique dans le style de [8] pourrait être définie afin d'avoir un système valide et complet, mais nous ne la présentons pas ici pour respecter la taille maximale autorisée de l'article. Elle est toutefois présentée dans [9].

Ont définit une *preuve* de φ à partir de l'ensemble de formules Σ, notée $\Sigma \vdash \varphi$, comme une séquence de formules telles que toute formule de cette séquence est soit un axiome, soit une formule de Σ, ou soit produite en appliquant une règle d'inférence sur des formules apparaissant précédemment dans la séquence.

Dans ce qui suit, $\bot$ représente toute formule qui est une contradiction et $\top$ représente toute formule qui est une tautologie.

3 Les réglementations et leurs propriétés

3.1 Règles et contraintes

Définition 3 (Règle)
Une règle est une formule de FOSDL de la forme $\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_n$ avec $n \geq 1$ telle que :

1. *$\forall \overrightarrow{x}$ représente $\forall x_1 \ldots \forall x_m$ où $\{x_1, \ldots, x_m\}$ est l'ensemble des variables libres apparaissant dans $l_1 \vee \ldots \vee l_n$*
2. *l_n est de la forme $O\varphi$ ou $\neg O\varphi$ où φ est un littéral objectif*
3. *$\forall i \in \{1, \ldots, n-1\}$, l_i est un littéral*
4. *si x est une variable de l_n, alors $\exists i \in \{1, \ldots, n-1\}$ tel que l_i est un littéral négatif qui contient la variable x*

D'après (1), les règles sont des formules fermées. D'après (2) et (3), les règles sont des conditionelles de la forme « si telle condition est vraie alors quelque chose est obligatoire (resp. permis ou interdit) ». En effet, $l_1 \vee \ldots \vee l_{n-1} \vee O\varphi$, est équivalent à $\neg l_1 \wedge \ldots \wedge \neg l_{n-1} \rightarrow O\varphi$. Finalement, (4) permet de restreindre les règles à des formules à champ restreint [2]. On peut remarquer que selon cette définition, seuls les littéraux objectifs peuvent être dans les modalités. Il n'y a donc pas possibilité d'imbriquer les modalités.

On écrira $\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee P\varphi$ pour $\{\forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee \neg O\varphi, \forall \overrightarrow{x}\ l_1 \vee \ldots \vee l_{n-1} \vee \neg O \neg \varphi\}$.

Définition 4 (Réglementation)
Une réglementation est un ensemble fini de règles.

Considérons l'exemple suivant qui sera utilisé tout le long de l'article.

Example 1
Nous considérons une réglementation qui règlemente le comportement d'un conducteur face à un feu de circulation. Le langage nécessaire est le suivant :

- *$|$, orange, rouge, voiture, camion, velo, A, B, B_1 et B_2 sont des constantes*
- *x, y, et z sont des variables*
- *Couleur(.) est un symbole de prédicat qui indique la couleur du feu*
- *Vehicule(.,.) est un symbole de prédicat qui représente le type du véhicule que le conducteur conduit*

2. Les formules à champ restreint forment un sous-ensemble décidable de l'ensemble des formules de FOL indépendantes du domaine. Il a été montré qu'elles sont les seules formules du premier ordre à avoir un sens en modélisation [7]

– *Devant*(.) *est un symbole de prédicat qui prend pour paramètres un conducteur et représente que ce conducteur est face au feu*
– *Stop*(.) *est un symbole de prédicat qui indique que le conducteur s'arrête au feu*
– *Avance*(.) *est un symbole de prédicat qui représente le fait que le conducteur avance et passe le feu*
– *Suit*(.,.) *est un symbole de prédicat représentant le fait qu'un conducteur suit un autre conducteur*

Considérons les trois règles suivantes :

(r_r) *« lorsqu'un automobiliste est face à un feu rouge, il doit s'arrêter »*

(r_o) *« lorsqu'un automobiliste est face à un feu orange, il peut s'arrêter »*

(r_g) *« lorsqu'un automobiliste set face à un feu vert, il ne doit pas s'arrêter »*

Ces règles sont modelisées par :

$(r_r) \forall x \, Vehicle(x, voiture) \wedge Color(rouge) \wedge IFO(x) \rightarrow OStop(x)$

$(r_o) \forall x \, Vehicle(x, voiture) \wedge Color(orange) \wedge IFO(x) \rightarrow PStop(x)$

$(r_g) \forall x \, Vehicle(x, voiture) \wedge Color(vert) \wedge IFO(x) \rightarrow FStop(x)$

Définition 5 (Contrainte)
Une contrainte est une formula objective fermée. Dans la suite, on notera C un ensemble de contraintes et on supposera que C est cohérent, i.e. $C \nvdash \bot$.

Afin de prendre en compte les contraintes dans le raisonnement, on considèrera une théorie de FOSDL dont les axiomes propres sont les formules de C . *On note $\vdash_C$ l'inférence dans cette théorie* et $Th_C(E)$ dénote l'ensemble des conséquences de E sous C, i.e $Th_C(E) = \{\varphi | E \vdash_C \varphi\}$.

Définition 6 (Etat du monde)
Un état du monde s est un ensemble complet de littéraux objectifs de base qui est cohérent avec C.

Intuitivement, les états du monde sont des représentations syntaxiques des interprétations de FOL. Pour des raisons de lisibilité, dans le suite *on omettra les littéraux négatifs lorsqu'on décrira un état du monde.*

Nous définisson maintenant les notions de cohérence et complétude des réglementations lorsqu'il existe des contraintes.

3.2 Cohérence des réglementations

Définition 7 (Cohérence sous contraintes)
Soit ρ une réglementation, C un ensemble de contraintes et s un état du monde cohérent. ρ est cohérente dans s sous C ssi $\rho, s \nvdash_C \bot$.

Cette définition est illustrée par les exemples suivants. Le premier exemple est un exemple basique d'une réglementation cohérente sous contraintes. Le second exemple montre que la définition met en évidence les conflits déontiques simples. Le troisième montre que les contraintes influencent la cohérence.

Example 2 (Exemple de base)
Considérons l'état du monde, la réglementation et les contraintes suivants :

– $s_0 = \{Vehicule(A, voiture), Couleur(rouge), IFO(A)\}$
– $\rho_0 = \{(r_r), (r_o), (r_g)\}$
– $C_0 = \{(oc)\}$ *avec* $(oc) = Couleur(rouge) \otimes Couleur(orange) \otimes Couleur(vert)$. *La contrainte* (oc) *signify que le feu a one et one seule couleur, rouge, vert ou orange.*

$\rho_0, s_0 \nvdash_{C_0} \bot$, *donc* ρ_0 *est cohérente dans* s_0 *sous* C_0.

Example 3 (Conflist déontiques simple)
Soient les deux réglementations :

– $\rho_1 = \{(r_r), (r'_r), (r_o), (r_g)\}$
avec $r'_r = \forall x \; Vehicule(x, voiture) \wedge Couleur(rouge) \wedge IFO(x) \rightarrow FStop(x)$
– $\rho_2 = \{(r_r), (r''_r), (r_o), (r_g)\}$
avec $r''_r = \forall x \; Vehicule(x, voiture) \wedge Couleur(rouge) \wedge IFO(x) \rightarrow PStop(x)$

$s_0, \rho_1 \vdash_{C_0} \bot$ *et* $s_0, \rho_2 \vdash_{C_0} \bot$. *Donc,* ρ_1 *et* ρ_2 *sont incohérentes dans* s_0 *sous* C_0.

Example 4 (Conflits avec les contraintes)

– $s_1 = \{Vehicule(A, voiture), Couleur(orange), IFO(A)\}$
– $\rho_3 = \{(r_r), (r'''_r)\}$. (r_r) *est la règle définie dans l'exemple 1 qui stipule qu'un automobiliste face à un feu rouge doit s'arrêter.* (r'''_r) *est one variante de* (r_r) *qui stipule qu'un automobiliste face à un feu rouge doit avancer. Elle est pourmalisée par :* $\forall x \; Vehicule(x, voiture) \wedge Couleur(rouge) \wedge Devant(x) \rightarrow OAvance(A)$.
– $\rho_4 = \{(r_o), (r'''_o)\}$. (r_o) *est la règle définie dans l'exemple 1 qui stipule qu'un automobiliste face à un feu orange peut s'arrêter. Et enfin,* (r'''_o) *est une variante de* (r_o) *où Stop est remplacé par Avance.*
– $C_0 = \{(oc)\}$
– $C_1 = \{(oc), (moveStop)\}$ *où* $(moveStop) = \forall x \neg(Stop(x) \wedge Avance(x))$

$s_1, \rho_3 \nvdash_{C_0} \bot$ *et* $s_1, \rho_3 \vdash_{C_1} \bot$ *:* ρ_3 *est cohérent dans* s_1 *sous* C_0 *mais incohérente sous* C_1. *Ceci signifie que sans la contrainte "il est impossible d'avancer et de s'arrêter en même temps", l'incohérence n'est pas capturée.*

$s_1, \rho_4 \nvdash_{C_0} \bot$ *et* $s_1, \rho_4 \nvdash_{C_1} \bot$ *:* ρ_4 *est cohérente dans* s_1 *sous* C_0 *et* C_1. *Dans ce dernier cas, il n'y a pas de* *contradiction parceque il existe une façon d'obér aux deux règles simultanément sans violer les contraintes.*

Ce dernier example montre que les constraints influencing la cohérence dune façon négative. En fait, la cohérence des réglementations peut être perdue s'il y a trop de contraintes.

3.3 Complétude des réglementations

Informellement, une réglementation est complète si et seulement si elle prescrit le comportement de tout agent dans toute situation. Cette propriété est importante. En effet, l'incomplétude signifie que certains comportements ne sont pas prescrits, ce qui n'est pas désirable, notamment dans des situations critiques où la sécurité est importante.

Remarquer que dans de nombreux cas, l'absence d'obligation et d'interdiction est interprétée par une permission implicite. Cette façon de compléter les réglementations nest pas satisfaisante car l'absence d'obligation et d'interdiction peut signifier une omission de la par du concepteur de la réglementation. Et, dans le cas de réglementations sur des systèmes critiques, de telles omissions doivent être capturées et ceci est l'objet de la propriété de complétude.

Ceci étant dit, il faut noter que la complétude totale n'a pas vraiment de sens. C'est pourquoi nous suggérons de définir une complétude partielle, restreinte quelques situations particulières et quelques comportement des agents particuliers.

Ceci conduit à la définition suivante :

Définition 8 (Complétude)
Soit C un ensemble de contraintes, s un état du monde et ρ une réglementation cohérente dans s sous C. Soit $\varphi(\overrightarrow{x})$ une formule objective et $\psi(\overrightarrow{x})$ une formule atomique, $\overrightarrow{x}$ representant les variables libres dans φ et $\psi(\overrightarrow{x})$ signifiant que les variables libres dans ψ sont un sous ensemble de $\overrightarrow{x}$. ρ est $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-complète *pour $\vdash_C$ dans s sis pour toute substitution de base χ telle que $s \vdash \varphi(\chi(\overrightarrow{x}))$:*

$$\rho, s \vdash_C O\psi(\chi(\overrightarrow{x})) \textit{ or}$$
$$\rho, s \vdash_C F\psi(\chi(\overrightarrow{x})) \textit{ or}$$
$$\rho, s \vdash_C P\psi(\chi(\overrightarrow{x}))$$

On peut se demander si FOSDL ne définit pas naturellement des réglementations complètes. En effete, par définition des operateurs P et F, si φ est une formule de base objective, alors $O\varphi$, $P\varphi$ et $F\varphi$ sont mutuellement exclusives et $O\varphi \vee P\varphi \vee F\varphi$ est une tautologie dans FOSDL. Donc, on peut conclure que FOSDL est suffisante pour éviter les réglementations incomplètes. Ceci n'est pas vrai. En effet, si ρ est une réglementation, s un état du monde et φ one formula objective, alors on peut déduire que $\rho, s \vdash O\varphi \vee P\varphi \vee F\varphi$ **mais** ceci ne signifie pas que l'on puisse déduire *soit* $\rho, s \vdash O\varphi$, *ou* $\rho, s \vdash P\varphi$ *ou* $\rho, s \vdash F\varphi$, ce qui nous intéresse précisément.

Les exemples suivants illustrent la notion de complétude. Le premier montre une réglementation complète et une réglementation incomplète. Le second montre que les contraintes influencent la notion de complétude.

Example 5 (Exemple de base)
Considérons à nouveau l'état du monde s_0, la réglementation ρ_0 et l'ensemble de contraintes $C_0 = \{(oc)\}$ come définis dans l'exemple 2. ρ_0 est cohérente dans s_0 sous C_0 et $\rho_0, s_0 \vdash O(Stop(A))$. Prenons $\varphi_0(x) \equiv Devant(x)$ et $\psi_0(x) \equiv Stop(x)$. $s_0 \vdash \varphi_0(A)$ et $\rho_0, s_0 \vdash_{C_0} O(\psi_0(A))$. Donc, ρ_0 est $(\varphi_0(x), \psi_0(x))$-complète pour $\vdash_{C_0}$ dans s_0.

Considérons maintenant l'état du monde $s_2 = \{Vehicule(A, camion), Devant(A), Couleur(rouge)\}$. ρ_0 est cohérente dans s_2 sous C_0. $s_2 \vdash \varphi_0(A)$ mais $\rho_0, s_2 \not\vdash_{C_0} O\psi_0(A)$, $\rho_0, s_2 \not\vdash_{C_0} P\psi_0(A)$ et $\rho_0, s_2 \not\vdash_{C_0} F\psi_0(A)$. Donc, ρ_0 est $(\varphi_0(x), \psi_0(x))$-incomplète pour $\vdash_{C_0}$ dans s_2. En fait, aucune règle de la réglementation ne peut être appliquée car le véhicule n'est pas une voiture mais un camion.

Example 6 (Utilisation des contraintes)
On considère l'état du monde s_0, la réglementation ρ_0 et un nouvel ensemble de contraintes $C_1 = \{(oc), (moveStop)\}$. On est maintenant intéressé dans l'action d'avancer : $\psi_1(x) \equiv Avance(x)$. Dans ce cars, la réglementation est (φ_0, ψ_1)-incomplète dans s_0 quand la seule contrainte est (oc) mais elle est (φ_0, ψ_1)-complète dans s_0 pour $\vdash_{C_1}$. En fait, $s_0, \rho_0 \vdash_{C_1} F\psi_1(A)$

La complétude est aussi influencée par les contraintes mais contrairement à la cohérence, cette influence est "positive". En fait, les contraintes peuvent rendre complète une réglementation incomplète.

4 Raisonner avec des réglementations incomplètes

Regardons maintenant les réglementations incomplètes. Nous pourrions (1) détecter les "trous" dans la réglementation et les signaler aux concepteur de la réglementation afin qu'il les "comble" ou (2) détecter les "trous" de la réglementation et appliquer des règles de complétion pour les faire disparaitre. La première solution est assez couteuse car le nombre de "trous" peut être élevé et donc les corriger peut être long. La seconde solution est celle qui existe dans les systèmes légaux et est appelé "the sealing legal principle". Toutefois, cette façon de compléter les réglementations est un peu systématique et n'offre pas de flexibilité. Nous adoptions dependent ce type d'approche tout en off rant one solution dans laquelle les règles de complétion sont "customisées".

Nous nous intéressons au fait que, ψ étant une formule objective, une réglementation déduise que ψ est obligatoire, interdit ou toléré. Par conséquent, si la réglementation est incomplète pour une formule objective ψ, alors elle ne sera complétée que par $O\psi$ ou $P\psi$, ou $F\psi$.

Afin de faire ce raisonnement, nous avons choisi d'utiliser la logique des défauts [15]. Plus précisément, nous adaptons la logique des défauts à la logique modale avec laquelle nous travaillons. Pour cela, nous proposons une nouvelle définition des défauts (très proche de la définition classique) et une nouvelle définition de la notion d'extensions qui prend en compte les contraintes.

4.1 Défauts dans $FOSDL$ et C-extensions

En nous basant sur la présentation de Besnard [2] on définit les défauts dans FOSDL par ceci :

Un défaut d dans FOSDL est une configuration $(P : J_1, \ldots, J_n/U)$ où $P, J_1, \ldots, J_n, U$ sont des formules fermées de FOSDL. P est appelée le *prérequis* de d, $J_1, \ldots, J_n$ la *justification* de d et U la *conséquence* de d. Une théorie des défauts $\Delta = (D, F)$ dans FOSDL est composée d'un ensemble de formules objectives fermées de FOSDL F (les faits) et d'un ensemble de défauts dans FOSDL.

On peut donner à une théorie des défauts (D, F) une *forme de surface* (D', F) par (1) tout défaut $d = (P(\overrightarrow{a}) : J_1(\overrightarrow{a}), \ldots, J_n(\overrightarrow{a})/U(\overrightarrow{a}))$ de D est tel que $P(\overrightarrow{x}) : J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x})/U(\overrightarrow{x}) \in D'$ et $\overrightarrow{a}$ est un terme de base, et (2) tout élément de D' est de la forme $(P(\overrightarrow{x}) : J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x})/U(\overrightarrow{x}))$ où $P(\overrightarrow{x})$, $J_1(\overrightarrow{x}), \ldots, J_n(\overrightarrow{x}), U(\overrightarrow{x})$ sont des formules de FOSDL avec des variables libres dans $\overrightarrow{x}$.

En utilisant les défauts classiques, on obtient des *extensions*, i.e. des ensembles de formules qui sont deduites de façon monotone et non-monotone à partir de F. Ici, nous tendons la notion d' extension en utilisant l'inférence $\vdash_C$ à la place de l'inférence classique $\vdash$. Les extensions obtenues avec $\vdash_C$ sont appelées des C-extensions.

Définition 9 (C-extensions)
Si $\Delta = (D, F)$ est une théorie des défauts dans FOSDL où les défauts ne containnient que des formules fermées de FOSDL, alors une C-extension de Δ set un ensemble de formules E tel que :

1. $E_0 = F$
2. *pour* $i \geq 0$, $E_{i+1} = Th_C(E_i) \cup \{U | \frac{P:J_1,\ldots,J_n}{U} \in D,$ $P \in E_i$ *et* $\neg J_1 \notin E, \neg J_2 \notin E, \ldots, \neg J_n \notin E\})$
3. $E = \bigcup_{i=0}^{\infty} E_i$

Reiter a montré dans [15] que si F est cohérente et si (D, F) a une extension, alors cette extension est cohérente. Il a aussi montré que toute théorie des défauts normaux et fermés a au moins une extension. On peut étendre ce résultat avec les C-extensions (une C-extension est incohérente si elle est l'ensemble de toutes les formules).

Proposition 1
Une théorie des défauts fermés de FOSDL (D, F) a uneC-extension inconsistante ssi $F \vdash_C \bot$.

Idée de preuve
Preuve évidente à partir de la définition 9.

Proposition 2
Soit D un ensemble de défauts fermés dans FOSDL. Soit F un ensemble de formules de FOSDL telles que $F \nvdash_C \bot$ alors la théorie des défauts $\Delta = (D, F)$ a au moins une C-extension.

Idée de preuve
Comme dans la preuve de Reiter, on construit une C-extension de Δ comme ceci : $E_0 = F$. Pour $i \geq 0$, soit T_i un ensemble maximal de formules fermées de FOSDL telles que (1) $E_i \cup T_i \nvdash_C \bot$ et (2) si $v \in T_i$ alors il existe un $\frac{P:U}{U}$, tel que v est U et $P \in E_i$. On définit $E_{i+1} = Th_C(E_i) \cup T_i$ et $E = \bigcup_{i=0}^{\infty} E_i$. On prouve que E est une $C-extension$ de Δ en montrant que $T_i = \{U | (P : U/U) \in D$, où $P \in E_i$ et $\neg U \notin E\}$.

4.2 Défauts pour compléter une réglementation

Pour compléter les réglementations, on utilise des défauts spécifiques et on définit une nouvelle relation d'inférence basée sur ces défauts.

Définition 10 (Ensemble de défauts)
Soit $E_F(\overrightarrow{x})$, $E_P(\overrightarrow{x})$ et $E_O(\overrightarrow{x})$ trois formules objectives telles que leur ensemble de variables libres respectifs sont dans $\overrightarrow{x}$. On définit l'ensemble de configurations $\mathcal{D}$ suivantes :

$$(DF_{\varphi,\psi}) \quad (\varphi(\overrightarrow{x}) \wedge E_F(\overrightarrow{x}) : F\psi(\overrightarrow{x})/F\psi(\overrightarrow{x}))$$
$$(DP_{\varphi,\psi}) \quad (\varphi(\overrightarrow{x}) \wedge E_P(\overrightarrow{x}) : P\psi(\overrightarrow{x})/P\psi(\overrightarrow{x}))$$
$$(DO_{\varphi,\psi}) \quad (\varphi(\overrightarrow{x}) \wedge E_O(\overrightarrow{x}) : O\psi(\overrightarrow{x})/O\psi(\overrightarrow{x}))$$

Définition 11 (Relation d'inférence $\vdash_C^*$)
Soit la théorie des défauts $\Delta_C = (\mathcal{D}, s \cup \rho)$ On définit une nouvelle relation d'inférence $\vdash_C^*$ par :
$s \cup \rho \vdash_C^* \gamma$ ssi γ appartient à toute C-extension de Δ_C.

Proposition 3
si ρ est cohérente dans s sous C, alors $\rho, s \nvdash_C^* \bot$.

Preuve
Trivial à partir de 1 et de la définition d'extension incohérente.

Ce résultat exprime le fait que si une réglementation est cohérente dans un état du monde sous des contraintes, alors, utiliser cette nouvelle inférence ne créera pas d'incohérences sous ces contraintes.

Comme dans la sous section précédente, on va définir la complétude pour cette nouvelle inference.

Définition 12 (Complétude pour $\vdash_C^*$)
*Soit s un état du monde et ρ une réglementation cohérente dans s sous C. Soit $\varphi(\overrightarrow{x})$ et $\psi(\overrightarrow{x})$ deux formules objectives définies dans 8. ρ est $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-*complète *dans s pour $\vdash_C^*$ ssi pour toute substitution de base χ telle que $s \vdash \varphi(\chi(\overrightarrow{x}))$, on a :*

$$\rho, s \vdash_C^* O\psi(\chi(\overrightarrow{x})) \textit{ ou}$$
$$\rho, s \vdash_C^* F\psi(\chi(\overrightarrow{x})) \textit{ ou}$$
$$\rho, s \vdash_C^* P\psi(\chi(\overrightarrow{x}))$$

Example 7 (Cas simple)
Dans cet example, on traite le cas où les contraintes ne concernent pas[3] la formule ψ pour laquelle on veut compléter. Considérons s_2, ρ_0, C_0, φ_0 et ψ_0 définis dans l'exemple 5.

ρ_0 est (φ_0, ψ_0)-incomplète dans s_2 pour $\vdash_{C_0}$ car aucune règle de ρ_0 ne s'applique pour A.

Prenons $E_O = E_P = E_F = \top$ Les extensions de Δ_{C_0} sont : $E_{1.F} = Th_{C_0}(s_2 \cup \rho_0 \cup \{F\psi_0(A)\})$ $E_{1.P} = Th_{C_0}(s_2 \cup \rho_0 \cup \{P\psi_0(A)\})$ $E_{1.O} = Th_{C_0}(s_2 \cup \rho_0 \cup \{O\psi_0(A)\})$

Aucune de ces trois formules $F\psi_0(A)$, $P\psi_0(A)$ et $O\psi_0(A)$ n'appartient à toute extension. Donc, aucune de ces formules ne peut être inférée avec $\vdash_{C_0}^*$ et la réglementation est incomplète pour $\vdash_{C_0}^*$.
Prenons maintenant $E_O = Couleur(rouge), E_F = Couleur(|), E_P = Couleur(orange)$. Seul le défaut DO_{φ_0,ψ_0} peut être appliqué. Donc, la seule extension est $E_{1.O} = Th_{C_0}(s_2 \cup \rho_0 \cup \{O\psi_0(A)\})$ et la réglementation ρ est maintenant complète pour $\vdash_{C_0}^*$.

Dans le dernier exemple, E_O, E_F et E_P sont des formules telles qu'une et une seule d'entre elles est vraie et la réglementation est donc complète pour $\vdash_C^*$. Nous généralisons ce résultat dans la proposition suivante.

Proposition 4
*Si $\vdash_C \forall \overrightarrow{x} E_0(\overrightarrow{x}) \otimes E_F(\overrightarrow{x}) \otimes E_P(\overrightarrow{x})$ et les contraintes C contraintes ne concernent pas le prédicat ψ alors pour toutes formules objectives $\varphi(\overrightarrow{x})$ et $\psi(\overrightarrow{x})$ définies comme dans 8, ρ est $(\varphi(\overrightarrow{x}), \psi(\overrightarrow{x}))$-*complète *dans s pour $\vdash_C^*$.*

Idée de preuve
On montre qu'il y a une unique extension E puis que E est complète pour $\vdash_C^*$.

3. I.e., les contraintes sont écrites sans utiliser le prédicat ψ.

Dans le cas où les contraintes concernent ψ, on ne peut pas garantir la complètude de la réglementation pour $\vdash_C^*$. Cependant, on peut construire des C-extensions telles que chaque extension représente une façon de compléter la réglementation, comme nous le montrons dans les propositions suivantes.

Proposition 5
Soient $E_F = E_O = E_P = \top$. Soient C un ensemble de contraintes, s un état du monde et ρ une réglementation cohérente dans s sous C. Soient φ et ψ deux prédicats définis comme dans 12. Si E est une C-extension de $\Delta_C = (\mathcal{D}, s \cup \rho)$ alors E est (φ, ψ)-complète pour $\vdash_C^*$.

Idée de preuve
Considérons une C-extension E de $\Delta_C = (\mathcal{D}, s \cup \rho)$. Supposons qu'elle est incomplète pour $\vdash_C^*$: $s \vdash_C \varphi(\overrightarrow{a})$ et $E \nvdash_C \Box\psi(\overrightarrow{a})$ pour tout $\Box \in \{O, P, F\}$. Cela signifie qu'aucun défaut de $\mathcal{D}$ a été appliqué. Par définition de la C-extension, cela implique que pour tout $\Box \in \{O, P, F\}$, $\neg\Box\psi(\overrightarrow{a}) \in E$. C'est impossible, car $O\psi(\overrightarrow{a}) \vee F\psi(\overrightarrow{a}) \vee O\psi(\overrightarrow{a})$ est une tautologie.

Les exemples suivants illustrent la dernière proposition et montrent comment la C-extension peut être utilisée pour obtenir toutes les façons de compléter une réglementation et de prendre en compte les contraintes.

Example 8
Considérons ρ_0, φ_0, ψ_0 comme précédemment et $s_3 = \{Vehicule(A, voiture), Vehicule(B, velo), Devant(A), Devant(B$ $Suit(B, A), Couleur(orange)\}$, $C_2 = \{(oc), (notWide)\} \cup F_1$ où $(notWide) = \forall x, y, Vehicule(x, voiture) \wedge Vehicule(x, velo) \wedge Devant(x)\ wedgeSuit(y, x) \rightarrow (Stop(x) \rightarrow Stop(y))$. Cela signifie que si un vélo suit une voiture et si la voiture s'arrête, alors le vélo ne peut que s'arrêter (car la route n'est pas assez large) et F_1 est l'ensemble suivant de faits persistants $\{Vehicule(A, voiture), Vehicule(B, velo), Devant(A), Suit(B, A$

Là encore, ρ_0 est (φ_0, ψ_0)-incomplète dans s_3 pour $\vdash_{C_2}$ car il n'y a pas de règles de ρ_0 qui s'applique pour B. Prenons $E_O = E_F = E_P = \top$.
Les défauts D_O et D_P peuvent s'appliquer, ce qui conduit à deux extensions : $E_1 = Th_{C_2}(s_4 \cup \rho \cup \{OStop(B)\})$, $E_2 = Th_{C_2}(s_4 \cup \rho \cup \{PStop(B)\})$. Chaque extension fournit une façon de compléter la réglementation.
Notons que dans cet exemple, si $E_F = \top$ et $E_O = E_P = \bot$, les défauts DO_{φ_0,ψ_0} et DP_{φ_0,ψ_0} ne peuvent plus s'appliquer, car leur prérequis est faux. La seule extensions est alors $E_3 = Th_{C_2}(s_3 \cup \rho_0)$ qui n'est pas complète pour $\vdash_{C_2}^*$. Ceci illustre le fait que la complétude ne peut pas être garantie dans le cas général.

Example 9

- $s_4 = \{ Vehicule(A, voiture), Vehicule(B_1, velo), Vehicule(B_2, velo), Suit(B_1, A), Suit(B_2, A), Devant(A), Devant(B_1), Devant(B_2), Couleur(rouge)\}$.
- $C_3 = \{(oc), (aLittleWider)\} \cup F_2$ *avec* $(aLittleWider) \equiv \forall x, y, z \ \ Vehicule(x, voiture) \wedge Vehicule(y, velo) \wedge Vehicule(z, velo) \wedge Devant(x) \wedge Suit(y, x) \wedge Suit(z, x) \rightarrow (Stop(x) \rightarrow (Stop(y) \vee Stop(z)))$ *qui signifie que lorsqu'une voiture est devant le feu tricolore que deux vélos la suivent, si la voiture s'arrête, alors au moins un des vélos ne peut pas passer (parce que la route est un peu plus large que dans l'exemple précédent, mais ne permet pas de faire passer les deux vélos) et* $F_2 = \{ Vehicule(A, voiture), Vehicule(B_1, velo), Vehicule(B_2, velo), Suit(B_1, A), Suit(B_2, A)\}$.
- ρ_0, φ_0 *et* ψ_0 *comme précédemment.*

ρ_0 *est* (φ_0, ψ_0)*-incomplète dans* s_3 *pour* $\vdash_{C_3}$ *car aucune règle de* ρ_0 *ne s'applique pour* B_1 *et* B_2.

Prenons $E_O = E_P = E_F = \top$. *Le lecteur pourra vérifier que chaque ensemble de la forme* $Th_{C_3}(s_3 \cup \rho_0 \cup \{\Box_1 Stop(B_1) \wedge \Box_2 Stop(B_2)\})$ *avec* $(\Box_1, \Box_2) \in \{(O, O), (O, P), (P, O), (P, P), (O, F), (F, O)\}$ *est une C-extension. La réglementation est encore incomplète pour* $\vdash^*_{C_3}$. *Cependant, chaque extension nous donne une façon de compléter la réglementation qui ne contredit pas les contraintes.*

5 Conclusion

Dans cet article, nous nous sommes intéressés au problème de l'analyse de la cohérence et de la complétude de réglementations qui peuvent exister dans une société d'agents afin de contraindre leur comportement.

Bien que ces notions aient déjà été étudiées dans [9], nous les étendons en prenant des contraintes générales en compte. Cela permet de simplifier le formalisme, car il n'y a plus besoin de partitionner le langage. Nous avons ensuite étudié l'influence de ces contraintes sur les propriétés de cohérence et de complétude : alors que les contraintes ont tendance à limiter la cohérence des réglementations, elles permettent parfois de les compléter. Nous avons également proposé une façon de compléter des réglementations incomplètes en utilisant une extension de la théorie des défauts de Reiter. Ce travail peut aussi être vu comme une proposition de solution au problème de prise en compte des contraintes décrit dans [10].

La notion de complétude développée ici est en fait une sorte de « complétude locale », au sens où nous demandons d'avoir $O(\psi(X))$, $P(\psi(X))$ ou $F(\psi(X))$ seulement pour un certain $\phi(X)$. Ceci ressemble beaucoup à la notion de complétude introduite dans le domaine des bases de données par [16] qui a noté que certaines contraintes exprimées dans une bases de données sont des règles indiquant ce que la base devrait savoir. Par exemple, la contrainte exprimant le fait que « tout employé a un numéro de téléphone, de fax ou une adresse mail » exprime le fait que pour tout employé connu par la base, cette dernière connaît son numéro de téléphone, son numéro de fax ou son adresse mail. Les défauts de Reiter peuvent être utilisés pour compléter une telle base de données en cas d'incomplétude. Par exemple, une des règles pourrait être que si la base de données ne contient aucune des informations requises pour un employé donné (ni numéro de téléphone, ni numéro de fax, ni adresse mail), alors on pourrait supposer que son numéro de téléphone est celui de son département. L'étude du lien existant entre la notion de complétude introduite ici et la notion de complétude locale constitue une extension intéressante de ce travail.

De plus, afin de pouvoir travailler avec des réglementations plus générales, notre travail pourrait être étendu. En particulier, nous pourrions considérons plus de notions, comme par exemple le temps et l'action. En fait, la notion de temps est très importante lorsque l'on traite des obligations et nous devrions considérer différents types de temps, au moins le temps de validité des normes et des échéances associés aux obligations. Notons également que dans la plupart des exemples de ce papier, les prédicats concernés par les opérateurs déontiques représentent des actions (avancer, stopper etc.). L'ajout d'un opérateur modal dynamique et/ou d'un opérateur temporel pourrait être intéressant. Nous obtiendrions alors qune logique multimodale avec un fort pouvoir expressif.

Références

[1] C. E. Alchourrón. Philosophical foundations of deontic logic and the logic of defeasible conditionals. In *Deontic logic in computer science : normative system specification.* John Wiley and Sons, 1994.

[2] P. Besnard. *An introduction to default logic.* Springer-Verlag, 1989.

[3] P. Bieber and F. Cuppens. Expression of confidentiality policies with deontic logic. In *Deontic logic in computer science : normative system specification*, pages 103–121. John Wiley and Sons, 1993.

[4] B. F. Chellas. *Modal logic, an introduction.* Cambridge University Press, 1980.

[5] L. Cholvy. Checking regulation consistency by using SOL-resolution. In *International Conference on Artificial Intelligence and Law*, pages 73–79, 1999.

[6] F. Cuppens and R. Demolombe. A modal logical framework for security policies. In *Lectures Notes in Artificial Intelligence*, volume 1325, page 1997. Springer, 1997.

[7] R. Demolombe. Syntactical characterization of a subset of domain independent formulas. *Journal of the Association for Computer Machinery*, 39(1) :71–94, 1982.

[8] M. Fitting and R. L. Mendelsohn. *First-order modal logic*. Kluwer Academic, 1999.

[9] C. Garion, S. Roussel, and L. Cholvy. A modal logic for reasoning on consistency and completeness of regulations. In Boella, Noriega, Pigozzi, and Verhagen, editors, *Normative Multi-Agent Systems*, Dagstuhl Seminar Proceedings. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[10] J. Hansen, G. Pigozzi, and L. W. N. van der Torre. Ten philosophical problems in deontic logic. In *Normative Multi-agent Systems*, 2007.

[11] O. Hart and J. Moore. Incomplete contracts and renegociation. *Econometrica*, 56(4) :755–785, 1988.

[12] S. Kanger. Law and logic. *Theoria*, (38), 1972.

[13] L. Lindahl. *Position and Change - a Study in Law and Logic*. Number 112 in Synthese Library. D. Reidel, 1977.

[14] P. Mc Namara. Deontic logic. Stanford Encyclopedia of Philosophy, 2006.

[15] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2), 1980.

[16] R. Reiter. What should a database know ? *Journal of Logic Programming*, 14(1,2) :127–153, 1992.

[17] L. M. M. Royakkers. Giving permission implies giving choice. In *International Workshop on Database and Expert Systems Applications*, pages 198–203, Los Alamitos, CA, USA, 1997. IEEE Computer Society.

[18] E. Vranes. The definition of ”norm conflict” in international law and legal theory. *The European Journal of International Law*, 17(2) :395–418, 2006.