Obligation change in Dependence Logic and Situation Calculus

Robert Demolombe[1] and Andreas Herzig[2]

[1] ONERA Toulouse
France
Robert.Demolombe@cert.fr
[2] IRIT-Université Paul Sabatier
France
herzig@irit.fr

Abstract. Obligation change raises the "frame problem" which is to characterise what obligations remain unchanged after an action has been performed. Many general solutions have been proposed but even if they are attractive from a thoretical point of view they have practical drawbacks.
In this paper simple solutions are proposed thanks to the restriction to obligations that take the form of modal literals. These solutions are presented in the framework of dependence logic and of situation calculus, and it is shown that they are based on the same intuitive idea. This idea is to express that we have a complete representation of actions and circumstances that can change an obligation.

1   Introduction

The problem of the characterisation of what remains to be true after the performance of an action is recognised as a difficult problem in the field of Articial Intelligence. This problem is usually called the "frame problem".

The same problem arises in the field of deontic logic if we want to characterise the set of obligations that persist after an action. It has some connections with deontic defeasibility but it is not the same problem (see [10, 1, 12, 9, 17]).

An interesting solution to the frame problem has been proposed by Reiter [11] in the framework of situation calculus for modelling the evolution of the world. Later on this solution has been extended to the evolution of beliefs about the world by Scherl and Levesque [16, 14]. This work has been extended to revision by Shapiro et al. in [15]. In [4] Demolombe has adapted their intuitive ideas to the evolution of obligations. However, this solution has practical drawbacks because it requires to assign to all the ideal situations an ideality level in the same way as Scherl and Levesque require the assignement of a plausibility level.

In this paper we investigate solutions to the frame problem which are less general, in the sense that we only consider facts that can be represented by literals, but are simpler to formalise and much easier to use for practical applications.

The first idea is to consider the dependence logic, which has been defined by Castilho, Herzig et al. in [2, 3] (section 2) and to extend it to obligations (section 3). The second idea is to extend the simple idea of successor state axioms in situation calculus to obligations about literals (section 4). At the end of the paper the two formalisations are compared and it is shown that they are based on the same intuitive ideas (section 5).

2   Dependence Logic

The dependence logic is a propositional modal logic with the two modal operators $\Box$ and $[\alpha]$. Sentences of the form $\Box(p)$ are read: p is true after any sequence of actions, and sentences of the form $[\alpha](p)$ are read: p is true after the action α.

For modelling an application domain the effects of the actions are defined by properties of the form:

$\Box(q \rightarrow [\alpha]p)$

For instance, in the typical example of the Yale Shooting Scenario we have:

$\Box(Loaded \rightarrow [shoot]\neg Alive)$

This intuitively means that after any sequence of actions, if the gun is loaded then after shooting the man is not alive.

In addition to the definition of the action effects we have a set of frame axioms of the form:

$\Box(\neg C \rightarrow (L \rightarrow [\alpha]L))$

where L is a literal and C is a formula of classical propositional logic.

In a metalanguage this axiom says that if we are not in the context C the truth value of L is independent of the action α. That is, there is a ternary relation between α, L and C, and the frame axioms could be represented in the metalanguage by this independence relation.

The problem is that for almost every applications the set of frame axioms is very large, because after an action most of the literals have the same truth value.

Then, it is easier to represent the dependence relation, which is the complement of the independence relation, than the independence relation itself. Let us call D the dependence relation, we suppose that D is finite. The fact that the tuple $< \alpha, P, C >$ is in D means that in the context C the truth value of the atom P may be changed by the action α. It is assumed that the dependence relation is **complete** in the sense that α may change the truth value of P only if there is a tuple $< \alpha, P, C >$ in D.

The logic which is based on this dependence relation is called LAPD [3]. It is formally defined as follows.

ATM is the set of atomic formulas of the language. We have $ATM = \{P, Q, \ldots\}$. LIT is the set of literals. ACT is the set of actions. We have $ACT = \{\alpha, \beta, \ldots\}$. $PFOR$ denotes the set of formulas of classical propositional logic.

The dependence relation is such that $D \subseteq ACT \times ATM \times PFOR$.

---

[3] LAPD abreviates Logic for Action and Plan with Dependence relation.

Semantics.

A model for the logic LAPD is a structure μ such that:
$$\mu = <W, \{R_\alpha \; : \; \alpha \in ACT\}, R_\Box, \tau>.$$

In μ:
- W is a set of possible worlds,
- $R_\Box$ and R_α are two accessibility relations which interpret $\Box$ and $[\alpha]$,
- τ is a function from ATM to 2^W; τ is extended as usual to the logical connectives.

The following constraints are imposed on μ:
- $R_\Box$ is reflexive and transitive,
- $R_\alpha \subseteq R_\Box$,
- if $wR_\alpha w'$ then
$$\forall P \in ATM \text{ if } \forall C \in PFOR(<\alpha, P, C> \in D \;\Rightarrow\; w \not\models C) \text{ then}$$
$$w \in \tau(P) \text{ iff } w' \in \tau(P).$$

The intuition of the last constraint is that if all the contexts C where α may influence P are false in w, then P has the same truth value in w and w'.

We adopt the notation:

$$Pre_D(\alpha, P) = \bigvee_{<\alpha, P, C> \in D} C$$

It is assumed that $Pre_D(\alpha, P) = \bot$ if there is no tuple in D of the form: $<\alpha, P, C>$.

Let us denote by $|L|$ the atom of the literal L. We have the property:

$$\models_{LAPD} \Box(\neg Pre_D(\alpha, |L|) \rightarrow (L \rightarrow [\alpha]L))$$

From the relation D we obtain the formula $Pre_D(\alpha, |L|)$, and from this property we have the corresponding frame axioms.

For instance, if $Pre_D(\alpha, P) = C$ we have the frame axioms:
$$\models_{LAPD} \Box(\neg C \rightarrow (P \rightarrow [\alpha]P))$$
$$\models_{LAPD} \Box(\neg C \rightarrow (\neg P \rightarrow [\alpha]\neg P))$$

Axiomatics.

The axiomatics of the LAPD logic is defined as follows:

- all the tautologies of the classical propositional logic,
- $[\alpha]$ obeys the schema K,
- $\Box$ obey the schemas K, T and 4,
- (I) $\Box p \rightarrow [\alpha]p$,
- $(Persist)$ $\neg Pre_D(\alpha, |L|) \rightarrow (L \rightarrow [\alpha]L)$,
- Modus Ponens and Necessitation for $\Box$ and $[\alpha]$.

It has been proved (see in the Annex) that this axiomatics is valid and complete.

Example.

We can see now how this logic can be applied to the Yale Shooting Scenario. The dependence relation D is $D = \{d1, d2, d3\}$, where we have:

(d1) $< load, Loaded, \top >$
(d2) $< shoot, Loaded, \top >$
(d3) $< shoot, Alive, \top >$

The set of effect laws is $LAW = \{1, 2, 3, 4\}$, where we have:

(1) $\Box[load]Loaded$

(2) $\Box[shoot]\neg Loaded$

(3) $\Box(Loaded \rightarrow [shoot]\neg Alive)$

(4) $\Box(\neg Loaded \wedge Alive \rightarrow [shoot]Alive)$

Let us assume that the current situation is represented by $KB = \{\neg Loaded,$ $Alive\}$.

Since there is no tuple in D of the form $< load, Alive, C >$ we have $Pre_D(load,$ $|Alive|) = \bot$. Then, from the schema $(Persist)$ we have the frame axiom:

(5) $Alive \rightarrow [load]Alive$

From (5) and KB we have:

(6) $[load]Alive$

From (1) and (T) we have:

(7) $[load]Loaded$

From the schema (I) and (3) we have:

(8) $[load](Loaded \rightarrow [shoot]\neg Alive)$

And from (7) and (8) we have:

(9) $[load][shoot]\neg Alive$

From (4) and (T) we also have:

(10) $\neg Loaded \wedge Alive \rightarrow [shoot]Alive$

Then, from KB we have:

(11) $[shoot]Alive$

If we apply the Necessitation rule to the frame axiom (5) we have:

(12) $[shoot](Alive \rightarrow [load]Alive)$

From (11) and (12) we have:

(13) $[shoot][load]Alive$

Finally we have:

$\vdash_{LAPD} KB \wedge LAW \rightarrow [load][shoot]\neg Alive$

$\vdash_{LAPD} KB \wedge LAW \rightarrow [shoot][load]Alive$

It is interesting to see how the property $Alive$ persists after the actions *shoot* and *load*.

3   Extension of Dependence Logic to Obligations

In the previous section we have seen how dependence logic provides us with a simple solution to the frame problem. This simplicity comes from the fact that the evolution of the world is described in terms of evolution of classical literals.

Here this approach is extended to the evolution of obligations, where this evolution is described in terms of modal literals.

We introduce the new modality Obg and sentences of the form $Obg(p)$ are read: it is obligatory that p. The new dependence logic extended to obligations is called $LAPDO$.

A modal literal has the form: $Obg(P)$, $\neg Obg(P)$, $Obg(\neg P)$ or $\neg Obg(\neg P)$, where $P \in ATM$. The set of modal literals is denoted by $LITM$.

If $LM \in LITM$ we denote by $|LM|$ the classical atom in LM. For instance, we have $|Obg(\neg P)| = P$.

To characterise the modal literals whose truth values may change after an action we define the dependence relation DO such that $DO \subseteq ACT \times ATM \times PFOR$.

The fact that a tuple $< \alpha, P, C >$ is in DO means that if C holds the action α may change the truth value of $Obg(P)$, $\neg Obg(P)$, $Obg(\neg P)$ or $\neg Obg(\neg P)$.

Semantics.

A model of the logic $LAPDO$ is a structure μ such that:
$$\mu = < W, \{R_\alpha \; : \; \alpha \in ACT\}, R_\square, R_{Obg}, \tau >.$$

In μ:

- W, $R_\square$, R_α and τ are defined like in $LAPD$ [4].
- R_{Obg} is an accessibility relation which interprets Obg and is reflexive.

The following constraints are imposed to μ:

- $R_\alpha \subseteq R_\square$ and $R_{Obg} \subseteq R_\square$,
- if $wR_\alpha w'$ then

  $\forall P \in ATM$ if $\forall C \in PFOR(< \alpha, P, C > \in D \;\Rightarrow\; w \not\models C)$ then
  $$w \in \tau(P) \text{ iff } w' \in \tau(P).$$

- if $wR_\alpha w'$ then

  $\forall P \in ATM$ if $\forall C \in PFOR(< \alpha, P, C > \in DO \;\Rightarrow\; w \not\models C)$ then
  $$w \in \tau(Obg(P)) \text{ iff } w' \in \tau(Obg(P)) \text{ and}$$
  $$w \in \tau(Obg(\neg P)) \text{ iff } w' \in \tau(Obg(\neg P)).$$

The last constraint on $LAPDO$ models means that if we are not in a context where the action α may change the truth values of the modal literals formed with P, then their truth values remain unchanged after α.

The constraint $R_{Obg} \subseteq R_\square$ requires some comments. Indeed, a consequence of this constraint is that we have $\models \square(p) \rightarrow Obg(p)$. Then, for example, from $\square(Loaded \rightarrow [shoot]\neg Alive)$ we can infer $Obg(Loaded \rightarrow [shoot]\neg Alive)$. This consequence may seem to be odd in a first approach.

In fact this is acceptable if the intuitive meaning of $Obg(p)$ is: p is true in all the ideal worlds, and if we accept that ideal worlds are a subset of the "real worlds". Here we call real world a world which satisfies all the properties that are necessarily true in a given application domain. In particular all the properties that define the effects of the actions must hold in a real world.

Why should we impose that the ideal worlds are a subset of the real worlds? Suppose, on the contrary, that there is an ideal world w which is not a real world. That means that in w there is a property of the domain which is not satisfied.

Let us consider, for example, the property: a person cannot be at two different places at the same time. Then, in w it could be the case that the same person is at two different places at the same time, and, from a normative point of view, it would be permitted for a person to be at two different places at the same time.

---

[4] The function τ is extended to obligations in a natural way. We have $\tau(Obg(p)) = \{ w \; : \; w \models Obg(p)\}$, and $w \models Obg(p)$ iff $wR_{Obg}w'$ implies $w' \models p$.

It would be very odd to define a regulation with such a permission. That is why it is imposed that ideal worlds are real worlds.

We adopt the notation:

$$Pre_{DO}(\alpha, P) = \bigvee_{<\alpha,P,C>\in DO} C$$

If $LM \in LITM$ we have the property:

$$\models_{LAPDO} \Box(\neg Pre_{DO}(\alpha, |LM|) \rightarrow (LM \rightarrow [\alpha]LM))$$

For example, if $Pre_{DO}(\alpha, P) = C$ we have:
$\models_{LAPDO} \Box(\neg C \rightarrow (Obg(P) \rightarrow [\alpha]Obg(P)))$
$\models_{LAPDO} \Box(\neg C \rightarrow (\neg Obg(P) \rightarrow [\alpha]\neg Obg(P)))$
$\models_{LAPDO} \Box(\neg C \rightarrow (Obg(\neg P) \rightarrow [\alpha]Obg(\neg P)))$
$\models_{LAPDO} \Box(\neg C \rightarrow (\neg Obg(\neg P) \rightarrow [\alpha]\neg Obg(\neg P)))$

Axiomatics.

The axiomatics of the $LAPDO$ logic is defined as follows:

- all the tautologies of the classical propositional logic,
- $[\alpha]$ obeys the schema K,
- $\Box$ obeys the schemas K, T and 4,
- Obg obeys the schemas K and D,
- (I) $\Box p \rightarrow [\alpha]p$,
- (O) $\Box p \rightarrow Obg(p)$,
- $(Persist)$ $\neg Pre_D(\alpha, |L|) \rightarrow (L \rightarrow [\alpha]L)$, if $L \in LIT$,
- $(Persist_O)$ $\neg Pre_{DO}(\alpha, |LM|) \rightarrow (LM \rightarrow [\alpha]LM)$, if $LM \in LITM$,
- Modus Ponens and Necessitation for $\Box$, $[\alpha]$ and Obg.

It has been proved that this logic is valid and complete (see in the Annex).

Example.

Let us take the example of the traffic lights to show how obligation change is formalised in $LAPDO$. We use the following notations [5]:

Red: the light is red.

Green: the light is green.

InCrossing: the car is crossing the crossroads.

For the actions we use the notations:

red: to switch the light to red.

green: to switch the light to green.

start.cr: to start to cross the crossroads.

end.cr: to end to cross the crossroads.

The relation D is $D = \{d1, d2, d3, d4, d5, d6\}$ where:

$(d1)$ $< red, Red, \top >$

$(d2)$ $< red, Green, \top >$

$(d3)$ $< green, Red, \top >$

$(d4)$ $< green, Green, \top >$

---

[5] As a matter of simplification we have ignored the case where the light is orange.

(d5) $< start.cr, InCrossing, \top >$

(d6) $< end.cr, InCrossing, \top >$

The relation DO is $DO = \{do1, do2\}$ where;

(do1) $< red, InCrossing, \top >$

(do2) $< green, InCrossing, \top >$

Note that $start.cr$ and $end.cr$ have no influence on obligations.

The set of effects laws is $LAW = \{l1, l2, l3, l4, l5, l6, l7\}$ where:

(l1) $\square[red]Red$

(l2) $\square[green]Green$

(l3) $\square[start.cr]InCrossing$

(l4) $\square[end.cr]\neg InCrossing$

(l5) $\square[red]Obg(\neg InCrossing)$

(l6) $\square[green]Perm(InCrossing)$

(l7) $\square\neg(Red \wedge Green)$

As usual $Perm(p)$ is an abreviation for $\neg Obg(\neg p)$. The current situation is represented by $KB = \{\neg InCrossing, Green, Perm(InCrossing)\}$.

From (l1) and (T) we have:

(1) $[red]Red$

From (Persist) we have:

$\neg InCrossing \rightarrow [red]\neg InCrossing$

Then, from KB we have:

(2) $[red]\neg InCrossing$

From (l5) and (T) we have:

(3) $[red]Obg(\neg InCrossing)$

Therefore from (1), (2) and (3) we have:

$\vdash_{LAPDO} LAW \wedge KB \rightarrow [red](Red \wedge \neg InCrossing \wedge Obg(\neg InCrossing))$

It is worth noting that $(Persist_O)$ does **not** allow to infer:

$Perm(InCrossing) \rightarrow [red]Perm(InCrossing)$

because we have the tuple $< red, InCrossing, \top >$ in DO. Then the permission $Perm(InCrossing)$ does not persist after the action red.

From (Persist) we have:

$Red \rightarrow [start.cr]Red$

By Necessitation we have:

$[red](Red \rightarrow [start.cr]Red)$

And from (1) we have:

(4) $[red][start.cr]Red$

From (l3) and (I) we have:

(5) $[red][start.cr]InCrossing$

From $(Persist_O)$ we have:

$Obg(\neg InCrossing) \rightarrow [start.cr]Obg(\neg InCrossing)$

And by Necessitation we have:

$[red](Obg(\neg InCrossing) \rightarrow [start.cr]Obg(\neg InCrossing))$

Then, from (3) we have:

(6) $[red][start.cr]Obg(\neg InCrossing)$

Therefore from (4), (5) and (6) we have

$\vdash_{LAPDO} LAW \wedge KB \rightarrow [red][start.cr](Red \wedge InCrossing \wedge Obg(\neg InCrossing))$

It can be shown in a similar way that we have:

$\vdash_{LAPDO} LAW \wedge KB \rightarrow [red][green][start.cr](Green \wedge InCrossing \wedge Perm(InCrossing))$

This example shows how the obligations about the fact $InCrossing$ are updated when the actions red and $green$ are performed.

4 A simple extension of Situation Calculus to obligation change

The situation calculus is a typed first order classical logic (except some limited fragments that are in the second order). The characteristic feature of this logic is that dynamic aspects are represented by the notion of situation, which can be quantified, and each predicate whose truth value may change when actions are performed has an argument of the type situation. These predicates are called fluents.

For instance, the fact that the light is red in the situation s is represented by $Red(s)$. A situation may be the initial situation S_0, or the situation obtained after performance of the action a from the situation s. This situation is represented by the term $do(a, s)$.

For example, the situation $do(start.cr, S_0)$ represents the situation where the car has crossed the crossroads, and $do(red, do(start.cr, S_0))$ represents the situation where the light has switched to red after the car has crossed the crossroads.

To solve the frame problem in a given application domain we have to define for each fluent the complete list of the actions and circumstances that cause the fluent to be true or that cause the fluent to be false.

For example, the action red causes the light to be red and the action $green$ causes the light not to be red. This is formally represented by:

$(S1)$ $\forall s \forall a(a = red \rightarrow Red(do(a, s)))$

$(S2)$ $\forall s \forall a(a = green \rightarrow \neg Red(do(a, s)))$

To represent the fact that there are no other action that cause Red or $\neg Red$ we have to add the properties:

$(C1)$ $\forall s \forall a(\neg Red(s) \wedge Red(do(a, s)) \rightarrow a = red)$

$(C2)$ $\forall s \forall a(Red(s) \wedge \neg Red(do(a, s)) \rightarrow a = green)$

It can be shown that (S1), (S2), (C1) and (C2) are logically equivalent to (SS1).

$(SS1)$ $\forall s \forall a(Red(do(a, s)) \leftrightarrow a = red \vee Red(s) \wedge \neg(a = green))$

In the same way we have:

$(SS2)$ $\forall s \forall a(Green(do(a, s)) \leftrightarrow a = green \vee Green(s) \wedge \neg(a = red))$

$(SS3)$ $\forall s \forall a(InCrossing(do(a, s)) \leftrightarrow a = start.cr \vee InCrossing(s) \wedge \neg(a = end.cr))$

Notice that from (SS1) and (SS2) it can be easily proved by induction that $\neg(Green(S_0) \wedge Red(S_0)) \rightarrow \forall s(\neg(Green(s) \wedge Red(s)))$.

If we assume that each action has a unique name, from (SS1) we have:

$\forall s(Red(do(start.cr, s)) \leftrightarrow Red(s))$

Its intuitive meaning is that the action *start.cr* does not change the fact that the color of the light is red. In other terms the status of *Red* persists after any action other than *red* and *green*. That gives a very simple solution to the frame problem.

In general, for each fluent we have to define a successor state axiom of the form:

$\forall s \forall a(p(do(a,s)) \leftrightarrow \Gamma^+(a,s) \vee p(s) \wedge \neg\Gamma^-(a,s))$

To avoid inconsistencies we have to impose the constraint:

$$\neg\exists s\exists a(\Gamma^+(a,s) \wedge \Gamma^-(a,s))$$

The solution to the frame problem is based on two key ideas: we define the evolution of the world by defining the evolution of each literal, and we assume that we have a complete knowledge of the causes of their evolution. The same ideas will be applied to the evolution of the obligations in the same way as Demolombe and Pozos did for the evolution of beliefs [6].

In a first step we define obligations in the same way as Scherl and Levesque have defined beliefs in the situation calculus.

We adopt the definition:

$$Obg(p,s) \overset{\text{def}}{=} \forall s'(O(s',s) \rightarrow p[s'])$$

where the arguments of the type situation have been removed in p, and they have been replaced by s' in $p[s']$. $O(s',s)$ is a classical predicate that plays the same role as an accessibility relation.

To define the successor state axioms for obligations the only difference is that modal literals correspond to four truth values, while classical literals correspond to two truth values.

For example, to define the evolution of the four modal literals formed with the atom *InCrossing* we have the properties:

$(OS1)$ $\forall s\forall a(\bot \rightarrow Obg(InCrossing, do(a,s)))$
$(OS2)$ $\forall s\forall a(a = red \rightarrow \neg Obg(InCrossing, do(a,s)))$
$(OS3)$ $\forall s\forall a(a = red \rightarrow Obg(\neg InCrossing, do(a,s)))$
$(OS4)$ $\forall s\forall a(a = green \rightarrow \neg Obg(\neg InCrossing, do(a,s)))$

And we have four properties to represent the fact that the causes of change are complete:

$(OC1)$ $\forall s\forall a(\neg Obg(InCrossing, s) \wedge Obg(InCrossing, do(a,s)) \rightarrow \bot)$
$(OC2)$ $\forall s\forall a(Obg(InCrossing, s) \wedge \neg Obg(InCrossing, do(a,s)) \rightarrow a = red)$
$(OC3)$ $\forall s\forall a(\neg Obg(\neg InCrossing, s) \wedge Obg(\neg InCrossing, do(a,s)) \rightarrow a = red)$
$(OC4)$ $\forall s\forall a(Obg(\neg InCrossing(s)) \wedge \neg Obg(\neg InCrossing, do(a,s)) \rightarrow a = green)$

It can be shown that (OS1)-(OS4) and (OC1)-(OC4) are logically equivalent to (OSS1) and (OSS2).

$(OSS1)$ $\forall s\forall a(Obg(InCrossing, do(a,s)) \leftrightarrow Obg(InCrossing, s) \wedge \neg(a = red))$

(OSS2) $\forall s \forall a(Obg(\neg InCrossing, do(a,s)) \leftrightarrow a = red \lor Obg(\neg InCrossing, s) \land \neg(a = green))$

Let us consider an initial situation defined by $KB = \{\neg InCrossing(S_0), Green(S_0), Perm(InCrossing, S_0)\}$.

From (SS1) we have:

(1) $Red(do(red, S_0))$

From (SS3) and KB we have:

(2) $\neg InCrossing(do(red, S_0))$

From (OSS2) we have:

(3) $Obg(\neg InCrossing, do(red, S_0))$

If we denote by AS the set of properties $AS = \{SS1, SS2, SS3, OSS1, OSS2\}$ we have:

$\vdash AS \land KB \rightarrow Red(do(red, S_0)) \land \neg InCrossing(do(red, S_0)) \land Obg(\neg InCrossing, do(red, S_0))$

Notice that in S_0 it is permitted to cross the crossroads while in $do(red, S_0)$ it is forbidden to cross. This shows that the action red requires obligation updating. We can also notice that the fact $\neg InCrossing$ persists after the action red.

From (SS1) and (1) we also have:

(4) $Red(do([red, start.cr], S_0))$ [6]

From (SS3) we have:

(5) $InCrossing(do([red, start.cr], S_0))$

From (OSS2) and (3) we have:

(6) $Obg(\neg InCrossing, do([red, start.cr], S_0))$

Therefore we have:

$\vdash AS \land KB \rightarrow Red(do([red, start.cr], S_0)) \land InCrossing(do([red, start.cr], S_0)) \land Obg(\neg InCrossing, do([red, start.cr], S_0))$

It can be shown in a similar way that we have:

$\vdash AS \land KB \rightarrow Green(do([red, green, start.cr], S_0)) \land InCrossing(do([red, green, start.cr], S_0)) \land Perm(InCrossing, do([red, green, start.cr], S_0))$

In general for each normative fluent we must define two successor state axioms for obligations of the form:

$\forall s \forall a(Obg(p, do(a,s)) \leftrightarrow \Gamma_1^+(a,s) \lor Obg(p,s) \land \neg \Gamma_1^-(a,s))$

$\forall s \forall a(Obg(\neg p, do(a,s)) \leftrightarrow \Gamma_2^+(a,s) \lor Obg(\neg p,s) \land \neg \Gamma_2^-(a,s))$

To guarantee the consistency of obligations we impose the constraints:

$$\neg \exists s \exists a(\Gamma_1^+(a,s) \land \Gamma_1^-(a,s))$$

$$\neg \exists s \exists a(\Gamma_2^+(a,s) \land \Gamma_2^-(a,s))$$

To satisfy the schema (D) we impose the constraint:

$$\neg \exists s \exists a(\Gamma_1^+(a,s) \land \Gamma_2^+(a,s))$$

Moreover, from (D) we have: $Obg(p, do(a,s)) \rightarrow \neg Obg(\neg p, do(a,s))$. In addition we have: $\Gamma_1^+(a,s) \rightarrow Obg(p, do(a,s))$. Then, we can infer: $\Gamma_1^+(a,s) \rightarrow$

---

[6] $do([red, start.cr], S_0)$ is an abreviation for $do(start.cr, do(red, S_0))$.

$\neg Obg(\neg p, do(a, s))$. Since $\Gamma_2^-(a, s))$ represent all the circumstances that cause $\neg Obg(\neg p, do(a, s))$ we must impose the constraint:

$$\forall s \forall a (\Gamma_1^+(a, s) \rightarrow \Gamma_2^-(a, s))$$

For a similar reason we impose the constraint:

$$\forall s \forall a (\Gamma_2^+(a, s) \rightarrow \Gamma_1^-(a, s))$$

5 Comparison between Situation Calculus and Dependence Logic

To analyse the links between the evolution of obligations expressed in the situation calculus or in the dependence logic, we shall consider a translation from situation calculus to a dynamic logic, and from this dynamic logic to dependence logic. Then, it is shown that consequences derived in dynamic logic correspond to the consequences derived in dependence logic.

5.1 From Situation Calculus to Dynamic Logic

In [5] Demolombe has presented a general method to translate situation calculus formulas into formulas of a dynamic logic.

As a matter of simplification we only consider here the translation of the successor state axioms for the obligations.

Without loss of generality it can be assumed that the Γ_is have the following form.

$\Gamma_1^+(a, s) \overset{\text{def}}{=} a = \alpha \wedge C_1^+(s)$

$\Gamma_1^-(a, s) \overset{\text{def}}{=} (a = \beta \wedge C_1^-(s)) \vee (a = \gamma \wedge C_2^+(s))$

$\Gamma_2^+(a, s) \overset{\text{def}}{=} a = \gamma \wedge C_2^+(s)$

$\Gamma_2^-(a, s) \overset{\text{def}}{=} (a = \delta \wedge C_2^-(s)) \vee (a = \alpha \wedge C_1^+(s))$

It is assumed that the C_is contain no symbol of the type action.

In Γ_1^- we have the subformula $a = \gamma \wedge C_2^+(s)$ because $a = \gamma \wedge C_2^+(s)$ implies $Obg(\neg p, do(a, s))$, and, since obligations should obey (D), $Obg(\neg p, do(a, s))$ implies $\neg Obg(p, do(a, s))$. Then, $a = \gamma \wedge C_2^+(s)$ causes $\neg Obg(p, do(a, s))$. We have $a = \alpha \wedge C_1^+(s)$ in Γ_2^- for a similar reason.

All the results would be the same if instead of Γ_1^+ we had:

$\Gamma_1^+(a, s) \overset{\text{def}}{=} (a = \alpha_1 \wedge C_{1,1}^+(s)) \vee \ldots \vee (a = \alpha_n \wedge C_{1,n}^+(s))$

The same comment holds for the other Γ_is.

Thanks to the unique name axioms we can easily check that the Γ_is satisfy all the constraints mentioned in the previous section.

Then, the properties that define the effects of the actions on the obligations, and the completion properties are:

(C1) $\forall s \forall a (a = \alpha \wedge C_1^+(s) \rightarrow Obg(p, do(a, s)))$

(C2) $\forall s \forall a ((a = \beta \wedge C_1^-(s)) \vee (a = \gamma \wedge C_2^+(s)) \rightarrow \neg Obg(p, do(a, s)))$

$(C3)$ $\forall s \forall a(a = \gamma \wedge C_2^+(s) \rightarrow Obg(\neg p, do(a, s)))$

$(C4)$ $\forall s \forall a((a = \delta \wedge C_2^-(s)) \vee (a = \alpha \wedge C_1^+(s)) \rightarrow \neg Obg(\neg p, do(a, s)))$

$(C5)$ $\forall s \forall a(\neg Obg(p, s) \wedge Obg(p, do(a, s)) \rightarrow a = \alpha \wedge C_1^+(s))$

$(C6)$ $\forall s \forall a(Obg(p, s) \wedge \neg Obg(p, do(a, s)) \rightarrow (a = \beta \wedge C_1^-(s)) \vee (a = \gamma \wedge C_2^+(s)))$

$(C7)$ $\forall s \forall a(\neg Obg(\neg p, s) \wedge Obg(\neg p, do(a, s)) \rightarrow a = \gamma \wedge C_2^+(s)$

$(C8)$ $\forall s \forall a(Obg(\neg p, s) \wedge \neg Obg(\neg p, do(a, s)) \rightarrow (a = \delta \wedge C_2^-(s)) \vee (a = \alpha \wedge C_1^+(s)))$

It is worth noting that the set of formulas (C1)-(C8) is logically equivalent to (C9) and (C10).

$(C9)$ $\forall s \forall a(Obg(p, do(a, s)) \leftrightarrow (a = \alpha \wedge C_1^+(s)) \vee Obg(p, s) \wedge \neg((a = \beta \wedge C_1^-(s)) \vee (a = \gamma \wedge C_2^+(s))))$

$(C10)$ $\forall s \forall a(Obg(\neg p, do(a, s)) \leftrightarrow (a = \gamma \wedge C_2^+(s)) \vee Obg(\neg p, s) \wedge \neg((a = \delta \wedge C_2^-(s)) \vee (a = \alpha \wedge C_1^+(s))))$

The translation of these properties into dynamic logic is based on the following property:

$\vdash Obg(p, do(a, s)) \leftrightarrow \forall s''(s'' = do(a, s) \rightarrow \forall s'(O(s', s'') \rightarrow p[s']))$

This property justifies the translation of $Obg(p, do(a, s))$ into $[a]Obg(p)$.

Formulas of the form $\forall s F(s)$ are translated in dynamic logic into $\Box F$, where all the arguments of the type situation have been removed from the fluents that occur in $F(s)$.

To translate formulas of the form $\forall a G(a)$ it is assumed that the quantification domain for the actions is the set of actions that occur in some formula to be translated, plus another distinct action ϵ.

Then, we assume that we have the following domain closure axiom for the actions:

$\forall a(a = \alpha \vee a = \beta \vee a = \gamma \vee a = \delta \vee a = \epsilon)$

From this axiom the translation of $\forall a G(a)$ is $G(\alpha) \wedge G(\beta) \wedge G(\gamma) \wedge G(\delta) \wedge G(\epsilon)$, which is equivalent to the set of formulas: $G(\alpha)$, $G(\beta)$, $G(\gamma)$, $G(\delta)$, $G(\epsilon)$.

Notice that it is not necessary to have several distinct actions $\epsilon_1, \ldots, \epsilon_n$ like ϵ. Indeed, in the evaluation of the conditions of the form: $\epsilon_i = \alpha$, $\epsilon_i = \beta$, $\epsilon_i = \gamma$ and $\epsilon_i = \delta$, we always get the result $\bot$ for every ϵ_i. Then, every ϵ_i would lead to a translated formula of the same form.

Finally the translation of the set of properties (C1)-(C8) leads to:

$(D1)$ $\Box(C_1^+ \rightarrow [\alpha]Obg(p))$

$(D2)$ $\Box(C_1^- \rightarrow [\beta]\neg Obg(p))$

$(D2')$ $\Box(C_2^+ \rightarrow [\gamma]\neg Obg(p))$

$(D3)$ $\Box(C_2^+ \rightarrow [\gamma]Obg(\neg p))$

$(D4)$ $\Box(C_2^- \rightarrow [\delta]\neg Obg(\neg p))$

$(D4')$ $\Box(C_1^+ \rightarrow [\alpha]\neg Obg(\neg p))$

$(D5)$ $\Box(\neg Obg(p) \wedge [\alpha]Obg(p) \rightarrow C_1^+)$

$(D6)$ $\Box(Obg(p) \wedge \neg[\beta]Obg(p) \rightarrow C_1^-)$

$(D6')$ $\Box(Obg(p) \wedge \neg[\gamma]Obg(p) \rightarrow C_2^+)$

$(D7)$ $\Box(\neg Obg(\neg p) \wedge [\gamma]Obg(\neg p) \rightarrow C_2^+)$

$(D8)$ $\Box(Obg(\neg p) \wedge [\delta]\neg Obg(\neg p) \rightarrow C_2^-)$

$(D8')$ $\Box(Obg(\neg p) \wedge [\alpha]\neg Obg(\neg p) \rightarrow C_1^+)$

Since in the situation calculus the actions are deterministic, in dynamic logic we must have the schema $\neg[\alpha]\neg p \rightarrow [\alpha]p$. Moreover, in the situation calculus every situation has a successor for any action. Then, we must also have the schema: $[\alpha]p \rightarrow \neg[\alpha]\neg p$. To sum it up, in the dynamic logic we must have the axiom schema (DD).

(DD) $[\alpha]p \leftrightarrow \neg[\alpha]\neg p$

5.2 From Dynamic Logic to Dependence Logic

We consider a propositional dynamic logic with the axiom schema (DD).

The effects of the actions are represented by the properties (D1-(D8').

From the properties (D1)-(D4') we know that the following tuples are in the dependence relation for obligations DO.

$(o1)$ $< \alpha, p, C_1^+ >$
$(o2)$ $< \beta, p, C_1^- >$
$(o3)$ $< \gamma, p, C_2^+ >$
$(o4)$ $< \delta, p, C_2^- >$

From the completion properties (D5)-(D8') we know that there is no other tuple in DO. Therefore we have: $DO = \{o1, o2, o3, o4\}$.

The set of formulas in LAW is (D1)-(D4').

To have the same properties as in the dynamic logic we add to the dependence logic the axiom schema (DD).

5.3 From Dependence Logic to Dynamic Logic

Let us consider a dependence logic with the axiom schema (DD).

Let us assume that in this dependence logic the effects of the actions are defined by the set of sentences in LAW, and the dependence relation for obligation is DO, and LAW and DO are the same as in the previous section.

We can show that in this dependence logic obligations change in the same way as in the previous dynamic logic.

Let us denote by Mp a modal literal of the form: $Obg(p)$, $\neg Obg(p)$, $Obg(\neg p)$ or $\neg Obg(\neg p)$. From the axiom schema $(Persist_O)$, and from the relation DO, we have the following frame axioms.

$(f1)$ $\neg C_1^+ \rightarrow (Mp \rightarrow [\alpha]Mp)$
$(f2)$ $\neg C_1^- \rightarrow (Mp \rightarrow [\beta]Mp)$
$(f3)$ $\neg C_2^+ \rightarrow (Mp \rightarrow [\gamma]Mp)$
$(f4)$ $\neg C_2^- \rightarrow (Mp \rightarrow [\delta]Mp)$

We can prove that in the dependence logic from (f1)-(f4) we can infer (D5)-(D8'). Since (D1)-(D4') are in the dependence logic and in the dynamic logic, in both logics we have (D1)-(D8'), and the evolution of obligations is the same.

For example, we can prove that (f1) implies (D5). Indeed, if (f1) is transformed in clausal form and if we apply Necessitation for the operator $\square$ we get: $\square(\neg Mp \vee [\alpha]Mp \vee C_1^+)$. Then, for $Mp = \neg Obg(p)$ we have: $\square(Obg(p) \vee [\alpha]\neg Obg(p) \vee C_1^+)$. Moreover, from the schema (DD) we have: $\neg[\alpha]Obg(p) \leftrightarrow$

$[\alpha]\neg Obg(p)$; then, we have: $\Box(Obg(p) \lor \neg[\alpha]Obg(p) \lor C_1^+)$, which is the clausal form of (D5).

In a similar way we can prove that (f2) implies (D6). Indeed, if (f2) is transformed in clausal form and if we apply Necessitation for the operator $\Box$ we have: $\Box(\neg Mp \lor [\beta]Mp \lor C_1^-)$. Then, for $Mp = Obg(p)$ we have: $\Box(\neg Obg(p) \lor [\beta]Obg(p) \lor C_1^-)$ which is the clausal form of (D6).

6 Conclusion

Two simple solutions to the frame problem for obligations have been presented in the framework of dependence logic and of situation calculus. These solutions are restricted to obligations that apply to classical literals, and obligations are given the semantics of standard deontic logic. As we can see by the traffic light example the solutions work for iterated obligation changes, too.

It has been shown that both frameworks lead to the same consequences for obligation change. At the intuitive level the two solutions are based on the same ideas. A technical difference is that dependence logic requires some kind of meta reasoning, while situation calculus deals with classical logic but modalities have to be represented by a predicate that plays the role of an accessibility relation. The similarity between intuitive ideas can be shown as follows.

Let us assume that the action ϵ has no influence on the obligations about the atom p. That means in the dependence logic that there is no tuple of the form $< \epsilon, p, C >$ in the dependence relation DO, and by meta reasoning we can infer $Pre_{DO}(\epsilon, p) = \bot$. Then, from $(Persist_O)$ we have the four frame axioms:

$Obg(p) \rightarrow [\epsilon]Obg(p)$
$\neg Obg(p) \rightarrow [\epsilon]\neg Obg(p)$
$Obg(\neg p) \rightarrow [\epsilon]Obg(\neg p)$
$\neg Obg(\neg p) \rightarrow [\epsilon]\neg Obg(\neg p)$

From the schema (DD) we have $\neg[\epsilon]\neg\phi \leftrightarrow [\epsilon]\phi$. Then the four frame axioms are equivalent to the two frame axioms:

$[\epsilon]Obg(p) \leftrightarrow Obg(p)$
$[\epsilon]Obg(\neg p) \leftrightarrow Obg(\neg p)$

In the situation calculus, since ϵ does not influence the obligations about p, ϵ is different from α, β, γ and δ. Then, from (C9) and (C10) we have:

$\forall s(Obg(p, do(\epsilon, s)) \leftrightarrow Obg(p, s))$
$\forall s(Obg(\neg p, do(\epsilon, s)) \leftrightarrow Obg(\neg p, s))$

We see that we obtain frame axioms that have the same semantics in both frameworks, the difference is just technical.

An important issue that deserves more work is the ramification problem, that is to integrate in these frameworks invariant constraints between obligations like, for example, in the situation calclus $\forall s(Obg(p, s) \rightarrow Obg(q, s))$. Solutions proposed by Lin and Reiter in [7] and McIllraith in [8] could be adapted to the case of modal literals.

References

1. A. Artosi, G. Governatori, and G. Sartor. Towards a computational treatment of deontic defeasibility. In M. A. Brown and J. Carmo, editors, *Deontic Logic, Agency and Normative Systems*, pages 27–46. Springer, 1996.

2. M. A. Castilho, O. Gasquet, and A. Herzig. Formalizing action and change in a modal logic I:the frame problem. *Journal of Logic and Computation*, 9(5), 1999.

3. M. A. Castilho, A. Herzig, and I. J.Varzinczak. It depends on the context! A decidable logic of actions and plans based on a ternary dependence relation. In *9th Int. Workshop on Non Monotonic Reasoning*, 2002.

4. R. Demolombe. From belief change to obligation change in the Situation Calculus. A preliminary study. In J. Horty and A.J.I. Jones, editor, *Six International Workshop on Deontic Logic in Computer Science*, 2002.

5. R. Demolombe. Belief change: from Situation Calculus to Modal Logic. In G. Brewka and P. Peppas, editor, *Proc. of the Workshop on Nonmonotonic Reasoning, Action and Change*, 2003.

6. R. Demolombe and M. P. Pozos-Parra. A simple and tractable extension of situation calculus to epistemic logic. In Z. W. Ras and S. Ohsuga, editors, *Proc. of 12th International Symposium ISMIS 2000*. Springer. LNAI 1932, 2000.

7. F. Lin and R. Reiter. State constraints revisited. *Journal of Logic and Computation*, 4:655–678, 1994.

8. S. McIllraith. A closed-form solution to the ramification problem (sometimes). In *Proc. of the IJCAI'97. Workshop on Non Monotonic Reasoning Action and Change*, 1997.

9. D. Nute. Norms, priorities and defeasibility. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 201–218. IOS Press, 1999.

10. H. Prakken. Two approaches of defeasible reasoning. In A.J.I. Jones and M. Sergot, editors, *2d International Workshop on Deontic Logic in Computer Science*, pages 281–295. Tano A.S., 1994.

11. R. Reiter. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, 1991.

12. L. Royakkers and F. Dignum. Defeasible reasoning with legal rules. In M. A. Brown and J. Carmo, editors, *Deontic Logic, Agency and Normative Systems*, pages 174–193. Springer, 1996.

13. Sahlqvist, H. Completeness and correspondence in the first and second order semantics for modal logics. In Stig Kanger, editor, *Proc. 3rd Scandinavian Logic Symposium 1973*, number 82 in Studies in Logic. North Holland, 1975.

14. R. Scherl and H.J. Levesque. Knowledge, action and the frame problem. *Artificial Intelligence*, 144:1–39, 2003.

15. S. Shapiro, M. Pagnuco, Y. Lespérance, and H. Levesque. Iterated belief change in the situation calculus. In *Proc. of the 7th Conference on Principles on Knowledge Representation and Reasoning (KR2000)*. Morgan Kaufman Publishers, 2000.

16. W. Snyder and C. Lynch. Goal directed strategies for paramodulation. In *Proc. 14th Int. Conf. on Rewriting Techniques and Applications (LNCS 488)*. Springer-Verlag, 1991.

17. L. W. N. van der Torre and Y. H. Tan. An update semantics for deontic reasoning. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems*, pages 73–92. IOS Press, 1999.

Annex

Soundness and completeness.

Soundness of $LAPDO$ can be proved as usual by proving that all the theorems are valid, and that the inference rules preserve validity. We prove completeness in several steps.

First we define the set of all instances of axioms $(Persist)$ and $(Persist_O)$:

$$Indep(D) = \{\neg Pre_D(\alpha, |L|) \rightarrow (L \rightarrow [\alpha]L) : \alpha \in ACT \text{ and } L \in LIT\}$$

$Indep(DO) =$

$$\{\neg Pre_{DO}(\alpha, |LM|) \rightarrow (LM \rightarrow [\alpha]LM) : \alpha \in ACT \text{ and } LM \in LITM\}$$

We abbreviate $Indep(D, DO) = Indep(D) \cup Indep(DO)$.

Let $LAPDO_0$ be the basic logic of dependence and obligations such that

$$D_0 = DO_0 = \{< \alpha, P, \top >: \alpha \in ACT \text{ and } P \in ATM\}$$

Lemma. If $\models_{LAPDO} p$ then $Indep(D, DO) \models_{LAPDO_0} p$.

This follows from the fact that the class of models of $LAPDO$ is just the same as the class of those models μ of $LAPDO_0$ where $Indep(D, DO)$ is true in μ. (A set of formulas is true in μ iff each of its elements is true in every possible world of μ.)

Now we restrict $Indep(D)$ and $Indep(DO)$ to the language of p:

$$Indep(D, DO, p) = Indep(D, DO) \cap lang(p)$$

where $lang(p)$ is the language of p, i.e. the set of formulas built from the actions and atoms appearing in p.

Lemma. If $Indep(D, DO) \models_{LAPDO_0} p$ then $Indep(D, DO, p) \models_{LAPDO_0} p$.

As $Indep(D, DO, p)$ is finite we can formulate the following.

Lemma. If $Indep(D, DO, p) \models_{LAPDO_0} p$ then

$$\models_{LAPDO_0} (\Box \bigwedge Indep(D, DO, p)) \rightarrow p.$$

This follows from the fact that $R_\Box$ contains the reflexive and transitive closure of the union of R_{Obg} and all the accessibility relations R_α.

In logic $LAPDO_0$ we have that $Pre_D(\alpha, P) = Pre_{DO}(\alpha, P) = \top$ for every α and p. Therefore axioms $(Persist)$ and $(Persist_O)$ are redundant in that logic and can be dropped. As the remaining axioms are standard ones, the following is guaranteed by Sahlqvist's completeness theorem [13].

Lemma. If $\models_{LAPDO_0} (\Box \bigwedge Indep(D, DO, p)) \rightarrow p$ then

$$\vdash_{LAPDO_0} (\Box \bigwedge Indep(D, DO, p)) \rightarrow p.$$

Finally we have:

Lemma. If $\vdash_{LAPDO_0} (\Box \bigwedge Indep(D, DO, p)) \rightarrow p$ then

$$Indep(D, DO) \vdash_{LAPDO_0} \rightarrow p.$$

and

Lemma. If $Indep(D, DO) \vdash_{LAPDO_0} \rightarrow p$ then $\vdash_{LAPDO} p$.

The latter is because the axioms of $LAPDO$ are those of $LAPDO_0$ plus axioms $(Persist)$ and $(Persist_O)$. The set $Indep(D, DO)$ collects all instances of the latter axioms.

Putting the preceding lemmas together we obtain that $\models_{LAPDO} p$ implies $\vdash_{LAPDO} p$. Hence our logic $LAPDO$ is complete.

It follows a fortiori that LAPD is complete, too.