INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Fondements de la révision dans un langage d'objets simple*

Isabelle Crampé, Jérôme Euzenat

**N° 3060**

Décembre 1996

THÈME 3

Rapport de recherche

ISSN 0249-6399

# Fondements de la révision dans un langage d'objets simple

Isabelle Crampé*, Jérôme Euzenat

Thème 3
Interaction homme-machine, images, données, connaissances

Rapport de recherche n° 3060 — Décembre 1996 — 46 pages

**Résumé :** L'ajout d'une connaissance dans une base de connaissance peut provoquer une inconsistance. La révision consiste alors à modifier la base pour la rendre consistante avec la dernière connaissance à ajouter. Résoudre ce problème est très utile dans l'assistance aux utilisateurs de bases de connaissance. Afin de poser les bases d'un tel mécanisme pour les objets, une représentation par objets minimale est formalisée. Elle est dotée de mécanismes d'inférence et d'une caractérisation syntaxique de l'inconsistance et de l'incohérence. La notion de base de connaissance révisée est définie sur ce langage. Un critère de minimalité, à la fois sémantique et syntaxique, permet de définir les bases révisées les plus proches de la base initiale.

**Mots-clé :** Révision — minimisation des modifications — représentation de connaissance par objets

*(Abstract: pto)*

Version du Mardi 10 Décembre 1996

* `Isabelle.Crampe@inrialpes.fr`

Unité de recherche INRIA Rhône-Alpes
655 avenue de l'Europe, 38330 Montbonnot-Saint Martin (France)
Téléphone: +33 (0)476 61 52 00 — Télécopie: +33 (0)476 61 52 52

# Foundations for revision in a simple object-based representation language

**Abstract :** The revision of a knowledge base, which is inconsistent with a particular addition, modifies the base such that it gets consistent with this addition. The solution of this problem is useful in order to assist users in modifying their knowledge bases and it shall be successfully applied in the context of objects. In order to provide the foundation of such a mechanism in object-based systems, a minimal object model is first formalised. The model is provided with deduction; inconsistency and incoherence are characterised syntactically. The notion of revision is defined for this language, and semantic and syntactic minimality criteria allow to define revised bases which are the closest to the initial knowledge base.

**Key-words :** Revision — change minimisation — object-based knowledge representation

Lors de l'évolution d'une base de connaissance suite à l'acquisition de nouvelles connaissances sur le domaine à modéliser, les ajouts successifs qui sont effectués peuvent provoquer des inconsistances. La révision est une opération qui consiste alors à modifier la base afin qu'elle puisse intégrer la dernière connaissance en restant consistante. La révision a été étudiée longuement dans le domaine de la logique [1,14,3]. L'objectif de ce travail est d'étudier la révision dans les représentations de connaissance par objets (RCO) en se basant sur leurs particularités.

En effet, l'opération de révision en logique reste problématique [11] car deux types de problèmes se posent pour la mettre en œuvre : des problèmes de complexité (l'algorithme de révision nécessite généralement un oracle décidant de la consistance) et des problèmes de complétude (les critères définis syntaxiquement ne correspondent pas aux critères sémantiques). C'est pourquoi, même si les travaux de la révision en logique sont immédiatement applicables aux objets en traduisant les représentations par objets dans un formalisme logique, utiliser les multiples entités (classes, objets, attributs) des modèles à objets et des règles d'inférence adaptées semble un bon moyen de définir une opération de révision spécifique aux RCO. D'autre part, la spécificité des modèles à objets conduit à une caractérisation particulière de l'inconsistance et à l'introduction de la notion d'incohérence pour refléter des problèmes spécifiques aux bases de connaissance à objets (existence de classes ne pouvant contenir d'instance), problèmes à régler également lors de la révision.

De nombreuses raisons poussent à utiliser la révision dans une base de connaissance [10]; notre ambition est de proposer à un utilisateur de choisir parmi les différentes bases révisées possibles [6], une base révisée étant un sous-ensemble de la base initiale consistant avec la dernière connaissance à introduire. Cette interaction avec l'utilisateur évite de décider a priori quelle est la «meilleure» base révisée, d'autant plus que l'ordre de préférence entre ces bases varie selon les utilisations de la base. Cependant, il est nécessaire de déterminer des critères de minimalité afin de ne pas conserver, parmi les bases révisées possibles, celles qui font des suppressions superflues.

Afin d'explorer le bien-fondé et la faisabilité d'un mécanisme de révision dans une RCO, cette étude commence par définir dans une première partie un langage de représentation de connaissance par objets relativement restreint. L'extension de ce langage, en cherchant à préserver les propriétés obtenues, fait partie des perspectives de ce travail. Définir un langage spécifique et ne pas considérer d'emblée une logique d'objets établie (par exemple, F-logic [9]) permet de se restreindre aux cas simples. L'éventuel inconvénient de l'approche serait que l'extension ne puisse se faire de manière continue vers d'autres systèmes plus élaborés, mais son intérêt est d'identifier les éventuels problèmes un par un plutôt que de se heurter de front aux problèmes d'indécidabilité et de complexité de ces logiques [11].

La sémantique du langage et des procédures rapides de déduction, inspirées des techniques couramment utilisées, sont données. Les propriétés de complétude du langage sont établies formellement et une forme normalisée de base de connaissance (nommée forme première), utilisée par les algorithmes et par la révision, est aussi introduite. La seconde partie définit la notion de base révisée et les critères de minimalité (syntaxique) et de proximité (sémantique) sur les bases révisées. Il est montré que les deux notions coïncident. Dans la troisième partie, la contribution de ce travail est alors replacée dans le cadre de son utilisation au sein d'un système réel, qui nécessite une vérification de la formulation; des extensions sont également envisagées.

Un index des notations utilisées se trouve à la fin du rapport.

# 1. Logique d'objets

Pour fixer le formalisme sur lequel la révision s'applique, le langage d'expression des objets est défini (§1.1). Il est doté d'une sémantique correspondant à ce qui est attendu des représentations par objets considérées et les notions d'inconsistance et d'incohérence sont définies (§1.2); l'interprétation élémentaire d'une base de connaissance, nécessaire dans les démonstrations, est alors donnée (§1.3). Un ensemble de règles d'inférence est proposé pour ce langage (§1.4); il permet d'assurer la complétude avec la sémantique et de donner une définition syntaxique d'inconsistance et d'incohérence (§1.5). Une forme normalisée des bases de connaissance, utilisée pour la révision, est ensuite présentée (§1.6).

## 1.1. Syntaxe

La syntaxe est découpée en deux parties : une première partie correspond à la grammaire donnant l'expression des entités de la base, alors que la seconde permet de considérer des types de données qui peuvent être ajoutés au système (entiers, booléens, ainsi que tout autre tout type défini par un type abstrait).

### Grammaire du langage

Le langage permet de décrire des classes en relation de spécialisation, des objets — instances de ces classes — et des attributs qui sont typés dans les classes par un domaine et qui sont valués dans les objets par une valeur. Le langage ne pose aucune contrainte particulière entre le domaine d'un attribut d'une classe et celui de ses surclasses. Les notions de consistance et de cohérence étudiées aux paragraphes §1.2 et §1.4 permettront de définir les bases correctes.

La figure 1 est la représentation graphique d'une base, contenant cinq classes ($c_0$, $c_1$, $c_2$, $c_3$, $c_4$), trois objets (ou instances) $i$, $i'$ et $i''$ rattachées respectivement à $c_2$, $c_3$ et $c_4$ et un seul attribut $a$.

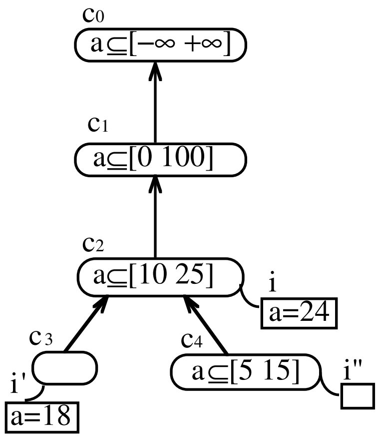


Figure 1 : exemple de base de connaissance représentée graphiquement.

**Définition 1 : grammaire**

<base_de_connaissance> ::= {<assertion>*}

<assertion> ::= <class_introduction> | <object_introduction> | <class_slot_restriction> | <object_slot_valuation>

<class_introduction> ::= <class_name> ≤ <class_name>

<object_introduction> ::= <object_name> ∈ <class_name>

<class_slot_restriction> ::= <class_name>**.**<slot_name> ⊆ <domain>

<object_slot_valuation> ::= <object_name>**.**<slot_name> **=** <value>

<class_name> | <object_name> | <slot_name> ::= <identifier>

<domain> et <value> sont définis ci-dessous.

De plus, une base de connaissance est un ensemble *fini* d'assertions et le graphe de la relation ≤ (appelée spécialisation) entre les <class_name> est sans circuit.

Selon la grammaire donnée, la base de connaissance de la figure 1 peut s'exprimer syntaxiquement par :

$B=\{c_1 \leq c_0\ ;\ c_2 \leq c_1\ ;\ c_3 \leq c_2\ ;\ c_4 \leq c_2\ ;\ i \in c_2\ ;\ i' \in c_3\ ;\ i'' \in c_4\ ;\ c_0.a \subseteq [-\infty\ +\infty]\ ;\ c_1.a \subseteq [0\ 100]\ ;\ c_2.a \subseteq [10\ 25]\ ;\ c_4.a \subseteq [5\ 15]\ ;\ i.a=24\ ;\ i'.a=18\}$.

## Notations

Dans la suite, $c$, $c'$, $c_1$,… dénoteront des classes, $i$, $i'$, $o$, $o'$ des objets, $a$, $a'$ des attributs, $v$, $v'$ des valeurs et $d$, $d'$ des domaines. Les assertions sont de l'une des quatre formes $c \leq c'$, $o \in c$, $c.a \subseteq d$ ou $o.a=v$. De plus, l'appartenance d'une assertion (par exemple $o \in c$) à une base $B$ est noté $(o \in c) \in B$.

**Opérations sur les types et les domaines**

Un type définit un ensemble de valeurs et des opérations sur celles-ci. Suivant les caractéristiques du type (ordonné, fini, etc.), les domaines destinés à représenter des sous-ensembles de valeurs du type ont une représentation différente (intervalles, vecteurs de bits, etc.) et les opérations applicables aux valeurs (égalité, comparaison, etc.) sont différentes. Pour chaque type $\tau$, on suppose qu'il existe un langage de valeurs $T_V^\tau$ et un langage de domaine $T_T^\tau$ disjoints de tous les autres langages utilisés ici.

Les opérations suivantes sur les domaines et les valeurs sont utilisées [5]; elles sont indicées par le type $\tau$ des domaines ou des objets concernés. Soient $d$, $d'$ des domaines et $v$, $v'$ des valeurs du type $\tau$.

$d \wedge_\tau d'$ : intersection des domaines de valeurs $d$ et $d'$;

$v :_\tau d$ ($\neg v :_\tau d$) : la valeur $v$ appartient (resp. n'appartient pas) au domaine $d$;

$d =_\tau d'$ ($d \neq_\tau d'$) : le domaine $d$ est égal au (resp. différent du) domaine $d'$;

$d \leq:_\tau d'$ ($d <:_\tau d'$) : le domaine $d$ est inclus (resp. strictement) dans le domaine $d'$;

$v \leq_\tau v'$ : la valeur $v$ est inférieure à la valeur $v'$; ($\tau$ étant ordonné).

$v =_\tau v'$ ($v \neq_\tau v'$) : la valeur $v$ est égale à (resp. différente de) la valeur $v'$.

$\perp_\tau$ est le symbole représentant le domaine vide du type $\tau$.

Le symbole $\tau$ sera omis là où il ne sera pas nécessaire. Dans le cadre de ce rapport, un sous-ensemble du modèle de connaissance est pris en compte en restreignant les valeurs aux entiers relatifs et les domaines aux intervalles d'entiers relatifs. Le seul type $\tau$ est donc entier :

<value> ::= <integer>

<domain> ::= [<bound$^-$> <bound$^+$>] / $\perp$

<bound$^-$> ::= <integer> / $-\infty$

<bound$^+$> ::= <integer> / $+\infty$

## 1.2. Sémantique

La sémantique est définie par une fonction d'interprétation des objets vers un domaine en tenant compte des types utilisés dans la syntaxe. Les notions de modèle, conséquence, inconsistance et incohérence sont définies ci-dessous. Afin de caractériser le domaine de la fonction d'interprétation, les ensembles de symboles manipulés sont tout d'abord identifiés.

**Définition 2 : ensembles d'entités**

Les ensembles suivants sont définis :

$T_C^B$ l'ensemble des classes $c$ introduites dans la base $B$ par une assertion de la forme $c \leq c'$ ou $c.a \subseteq d$.

$T_O^B$ l'ensemble des objets $o$ introduits dans la base $B$ par une assertion de la forme $o.a = v$ ou $o \in c$.

$T_A^B$ l'ensemble des attributs $a$ introduits dans la base $B$ par une assertion de la forme $o.a = v$ ou $c.a \subseteq d$.

$T_V^B$ l'ensemble des valeurs des types présents dans la base $B$ (ici les entiers).

$T_T^B$ l'ensemble des domaines des types présents dans la base $B$ (ici les intervalles d'entiers).

L'interprétation se définit pour les objets de la base, mais aussi pour les valeurs et domaines.

**Définition 3 : interprétation de type**

Soit un type $\tau$, une interprétation des expressions (domaines et valeurs) du type $\tau$ est une fonction $I_\tau$: $T_V^B \rightarrow \tau$ et de $T_T^B \rightarrow 2^\tau$ satisfaisant les condition suivantes :

| | | | | | |
|---|---|---|---|---|---|
| $I_\tau[\bot_\tau]$ | $=$ | $\emptyset$ | $I_\tau[d \wedge_\tau d']$ | $=$ | $I_\tau[d] \cap I_\tau[d']$ |
| $d <:_\tau d'$ | $\Leftrightarrow$ | $I_\tau[d] \subset I_\tau[d']$ | $d =_\tau d'$ | $\Leftrightarrow$ | $I_\tau[d] = I_\tau[d']$ |
| $v :_\tau d$ | $\Leftrightarrow$ | $I_\tau[v] \in I_\tau[d]$ | $\neg v :_\tau d$ | $\Leftrightarrow$ | $I_\tau[v] \notin I_\tau[d]$ |
| $v \neq v'$ | $\Leftrightarrow$ | $I_\tau[v] \neq I_\tau[v']$ | $v = v'$ | $\Leftrightarrow$ | $I_\tau[v] = I_\tau[v']$. |

Dans le cadre de ce rapport, un domaine est interprété par un intervalle d'entiers et une valeur par un entier; la fonction d'interprétation de type sera notée $I$. L'interprétation des domaines peut alors être précisée :

$I[[v\ \ v']] \quad = \quad \{v'' \in \tau, \text{ tel que } I[v] \leq_\tau v'' \leq_\tau I[v']\}$ (avec $\forall v \in \tau$, $I[-\infty] \leq_\tau v \leq_\tau I[+\infty]$)

La fonction d'interprétation de type sera considérée ici comme fixée (il n'en existe qu'une, donnée par l'implémentation, qui correspond aux intervalles sur les entiers relatifs). Il n'en sera donc plus question. La fonction d'interprétation va permettre de définir la relation entre la base et un domaine particulier.

**Définition 4 : interprétation des entités d'une base**

Soit une base de connaissance $B$ et un domaine $D$ quelconque. Une interprétation est un couple $<D, I_B>$, $I_B$ étant appelée fonction d'interprétation :

$I_B$: $T_O^B \rightarrow D$ injective sur $T_O^B$ (hypothèse de nom unique pour les objets)

$I_B$: $T_C^B \rightarrow 2^D$

$I_B$: $T_A^B \rightarrow (D \rightarrow \tau)$ $I_B[a]$ est une fonction totale

$I_B = I$ (l'interprétation du type $\tau$) sur $T_V^B$ et sur $T_T^B$.

Une interprétation d'une base associe donc à un objet, un élément du domaine, à une classe, un ensemble de tels éléments et à un attribut, une fonction du domaine vers le type.

Un certain nombre d'observations peuvent être faites sur la sémantique du langage proposé :

(1) La multi-spécialisation (la possibilité pour une classe d'être sous-classe de plusieurs classes incomparables) est autorisée. Plus généralement, aucune structure n'est imposée au graphe de la relation ≤ (hors l'absence de circuit).

(2) La multi-instanciation (la possibilité pour un objet d'être attaché à plusieurs classes incomparables) est autorisée.

(3) La multi-valuation (c'est-à-dire la faculté d'avoir plusieurs valeurs dans un attribut — comme c'est le cas d'une relation en général —, par opposition à avoir une collection de valeurs) est interdite par la sémantique des attributs qui sont des fonctions (à valeurs dans $\tau$ ici).

(4) La valeur d'attribut d'un objet ne peut être un autre objet (c'est la limitation principale de ce langage).

**Notations**

Pour un attribut $a$ et une classe $c$, $I_B[a|c]$ est l'image de $I_B[c]$ (l'ensemble des éléments de l'interprétation de cette classe) par $I_B[a]$ et $I_B[a|c] \subseteq I_B[d]$ exprime que $I_B[a|c]$ est une fonction de $I_B[c]$ dans $I_B[d]$. L'indice $B$ est omis lorsqu'il n'y a pas d'ambiguïté.

La définition de la satisfaisabilité d'une assertion (le fait qu'elle soit considérée comme vraie dans une interprétation) et de modèle (interprétation pour laquelle toutes les assertions de la base sont considérées comme vraies) sont données ci-dessous.

**Définition 5 : satisfaction d'une assertion**

Une interprétation $<D,I>$ satisfait une assertion $\delta$ est noté $\models_{<D,I>} \delta$ et

| | | |
|---|---|---|
| $\models_{<D,I>} c \leq c'$ | ssi | $I[c] \subseteq I[c']$ |
| $\models_{<D,I>} c.a \subseteq d$ | ssi | $I[a|c] \subseteq I[d]$ |
| $\models_{<D,I>} o \in c$ | ssi | $I[o] \in I[c]$ |
| $\models_{<D,I>} o.a = v$ | ssi | $I[a](I[o]) = I[v]$ |

**Définition 6 : modèle**

$M$=$<D, I>$ est un modèle d'une $B$, noté $\models_M B$, ssi $M$ satisfait toutes les assertions de la base $B$.

Ces définitions introduisent des contraintes sur la fonction d'interprétation d'un modèle d'une base qui correspondent à ce qui est attendu d'un modèle à objets :

(1) L'interprétation d'une classe doit être incluse dans l'interprétation de chacune de ses surclasses et dans l'ensemble d'éléments du domaine qui satisfont les restrictions de ses attributs. Si aucune contrainte n'est posée sur la classe (par

exemple pour la racine d'une hiérarchie), son interprétation est simplement incluse dans $D$.

(2) L'interprétation d'un objet doit appartenir à l'interprétation de chacune des classes à laquelle il est attaché et à l'ensemble des objets qui satisfont les contraintes posées sur la valeur de ses attributs.

(3) Le co-domaine de la restriction de l'interprétation d'un attribut à une classe doit être inclus dans celui de ce même attribut pour les surclasses et dans l'interprétation des domaines qui restreignent l'attribut pour cette classe particulière. Si aucune contrainte n'est posée sur un attribut, son domaine est simplement inclus dans $\tau$.

(4) L'interprétation d'un attribut d'un objet doit appartenir à l'interprétation de l'attribut de ses classes et aux interprétations des valeurs données pour cet attribut.

Les contraintes sur chacun des éléments sont synthétisées dans la proposition suivante :

**Proposition 1 :**

Si $M$=<$D$,$I_B$> est un modèle de $B$, alors :

$$I_B[c] \subseteq D \cap \bigcap_{(c \leq c') \in B} I_B[c'] \cap \bigcap_{(c.a \subseteq d) \in B} \{i \in D; I_B[a](i) \in I_B[d]\}$$

$$I_B[o] \in \bigcap_{(o \in c) \in B} I_B[c] \cap \bigcap_{(o.a = v) \in B} \{i \in D; I_B[a](i) = I_B[v]\}$$

Les fonctions dans $I[T_A^B]$ sont covariantes, c'est-à-dire que si $I[c] \subseteq I[c']$ alors $I[a|c] \subseteq I[a|c']$.

$$I_B[a|c] \subseteq \tau \cap \bigcap_{(c \leq c') \in B} I_B[a|c'] \cap \bigcap_{(c.a \subseteq d) \in B} I_B[d]$$

$$I_B[a](I_B[o]) \in \bigcap_{(o \in c) \in B} I_B[a|c] \cap \bigcap_{(o.a = v) \in B} \{I_B[v]\}$$

Les notions classiques de conséquence (ce qui est vrai dans tous les modèles d'une base particulière) et de base inconsistante (les bases n'ayant pas de modèle) sont maintenant introduites.

**Définition 7 : conséquence**

Soient $B$ une base et $\delta$ une assertion, $\delta$ est une conséquence de $B$, noté $B \models \delta$, ssi :

$$\forall \text{M modèle de } B, \models_{\text{M}} \delta.$$

La négation de la conséquence est notée $\not\models$.

**Définition 8 : base inconsistante**

Une base $B$ est inconsistante ssi il n'existe pas de modèle de $B$.

Par exemple, la base de la figure 1 à laquelle est simplement ajoutée l'assertion $(i''.a=7)$ est inconsistante (figure 2). En effet, $(i''.a=7) \in B$, $(i'' \in c_4) \in B$, $(c_2.a \subseteq [10\ 25]) \in B$ et

$(c_4 \leq c_2) \in B$. Tous les modèles doivent donc satisfaire : $I[a|c_4] \subseteq [10\ 15]$ d'où $I[a](I[i'']) \in \{7\} \cap [10\ 15]$, ce qui est impossible.

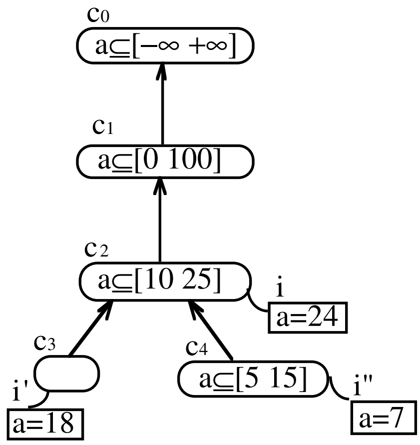


Figure 2 : exemple de base inconsistante.

Cette notion d'inconsistance n'est pas assez forte pour recouvrir la notion voisine d'incohérence. La notion d'incohérence permet de détecter les bases dans lesquelles il existe des classes qui ne peuvent avoir d'instance dans aucun des modèles, comme celle de la figure 3.

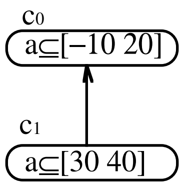


Figure 3 : exemple de base consistante mais incohérente

Ces classes représentent des potentiels d'inconsistance vu que les assertions de rattachement d'un objet à ces classes sont inconsistantes.

**Définition 9 : classe incohérente**

Une classe *c* est incohérente dans une base consistante *B* ssi son interprétation est vide dans tout modèle de *B*

**Définition 10 : base incohérente**

Une base consistante *B* est incohérente ssi elle contient une classe incohérente. Au contraire, une base *B* est cohérente ssi elle ne contient aucune classe incohérente.

Les base acceptables sont des bases qui ne contiennent pas de connaissances contradictoires, ni de classes incohérentes :

**Définition 11 : base acceptable, base inacceptable**

Une base *B* est acceptable ssi elle est consistante et cohérente. Dans le cas contraire, elle est inacceptable.

**Définition 12 : assertion inacceptable dans une base**

Une assertion $\delta$ est inacceptable dans une base $B$ ssi la base $B \cup \{\delta\}$ est inacceptable.

La notion de chaîne entre classes permet de simplifier les écritures, en particulier dans les démonstrations.

**Définition 13 : chaîne entre classes**

Une chaîne d'une classe $c$ à une classe $c'$ dans une base $B$ est un ensemble de classes $c_1, \ldots c_n$ telles que $c_1=c$, $c_n=c'$ et $\forall i \in [1..n\text{-}1]$, $(c_i \leq c_{i+1}) \in B$.

L'interprétation élémentaire, étudiée dans le paragraphe suivant, est introduite dans le seul but de démontrer certains des théorèmes qui suivent (la complétude faible notamment). Cette interprétation est l'interprétation la plus intuitive pour une base donnée et permet notamment de distinguer les classes incohérentes des classes cohérentes auxquelles aucun objet n'est attaché.

## 1.3. Interprétation élémentaire

L'interprétation élémentaire se définit en étendant le noyau d'interprétation. C'est un modèle de la base si la base est consistante et elle permet de donner une nouvelle caractérisation de base inconsistante ou incohérente.

Le noyau d'interprétation regroupe ce qui est vérifié par tous les modèles :

à un objet de la base est associé un objet du domaine;

à une classe est associé l'ensemble des instances attachés à cette classe ou à l'une de ses sous-classes;

à un attribut d'objet est associée l'interprétation de sa valeur dans la base (s'il possède une telle valeur);

à un attribut de classe est associé le domaine résultant de l'intersection des différentes contraintes posées (par ses surclasses ou par les restrictions de la classe elle-même).

**Définition 14 : noyau d'interprétation**

$K_B$ est un noyau d'interprétation d'une base $B$ sur un domaine $D_K$ si

$K_B$: $T_O \rightarrow D_K$ est une bijection

$K_B$: $T_C \rightarrow 2^{D_K}$

$$c \rightarrow \{K_B(o) \,;\, (o \in c) \in B\} \cup \bigcup_{(c' \leq c) \in B} K_B(c')$$

$K_B$: $T_A \rightarrow (D_K \rightarrow \tau)$ est une fonction partielle

$$K_B(a)\text{: } (K_B(c) \rightarrow \tau \cap \bigcap_{(c.a \subseteq d_i) \in B} I[d_i] \cap \bigcap_{(c \leq c') \in B} K_B(a|c'))$$

$K_B(a)(K_B(o)) = I[v]$ si $(o.a=v) \in B$.

**Remarques :**

- $K_B$ ne constitue pas nécessairement une interprétation (car $K_B(a)$ n'est pas définie sur tout l'ensemble $\{K_B(o), o \in T_O\}$), mais la notation $K_B(a|c)$ sera utilisée comme précédemment.
- Le noyau d'interprétation dépend de la bijection initiale de $T_O$ vers $D_K$. L'ensemble des noyaux d'interprétation constitue une classe de fonctions équivalentes à un isomorphisme sur $D_K$ près. On considérera dorénavant un unique $K_B$.

Pour définir la fonction d'interprétation, il reste alors à :

- interpréter les classes auxquelles aucun objet n'est attaché en leur associant un objet arbitraire (noté $o_c$ dans la définition suivante) satisfaisant toutes les contraintes posées sur la classe (ceci afin de pouvoir distinguer les cas de classe incohérente de ceux de classes sans instances)
- donner une valeur à tous les attributs des instances qui n'ont pas de valeur définie dans la base, en respectant les contraintes posées par les classes d'appartenance de l'instance.

**Définition 15 : interprétation élémentaire**

$<D,\hat{I}>$ est une interprétation élémentaire de $B$ si $\hat{I}$ est une fonction qui étend $K_B$ un noyau d'interprétation sur $D_K$, de la manière suivante :

$\hat{I}(o) = K_B(o)$

$\hat{I}(a|c) = K_B(a|c)$

$$\hat{I}(a)(\hat{I}(o)) \begin{cases} = K_B(a)(K_B(o)) & \text{si défini} \\ \in \bigcap_{(o \in c) \in B} \hat{I}(a|c) & \text{sinon} \end{cases}$$

$$\hat{I}(c) = \begin{cases} \emptyset & \text{si } \exists a,\ K_B(a|c)=\emptyset \\ \{o_c\} & \text{si } K_B(c)=\emptyset, \forall a\ K_B(a|c)\neq\emptyset \text{ et } \forall c',\ (c' \leq c) \notin B \\ \{K_B(o) \text{ tel que } \{(o \in c) \in B\} \cup \bigcup_{(c' \leq c) \in B} \hat{I}(c') & \text{sinon} \end{cases}$$

$D$ est l'ensemble $D_K$ augmenté de l'ensemble des objets $o_c$ nécessaires pour la définition.

Il existe en général plusieurs interprétations élémentaires, car les valeurs des attributs des objets (et en particulier des objets introduits dans la définition et notés $o_c$) ne sont pas nécessairement déterminées. Par abus de notation, toute interprétation élémentaire sera notée $<D,\hat{I}>$.

Les trois lemmes suivants permettent de localiser aisément les cas d'inconsistance et ainsi de préciser des particularités de bases consistantes, ce qui sera utile pour démontrer que l'interprétation élémentaire est modèle d'une base consistante.

**Proposition 2 :**

S'il existe $o \in T_O$ tel que $\bigcap_{(o \in c) \in B} K_B(a|c) = \emptyset$, alors la base est inconsistante.

*démonstration.* Si $\bigcap_{(o \in c) \in B} K_B(a|c) = \emptyset$, alors d'après la proposition 1, tout modèle de $B$ doit satisfaire $\bigcap_{(o \in c) \in B} I(a|c) = \emptyset$ et donc $I[a](I[o]) \in \emptyset$, ce qui rend impossible l'existence d'un modèle. ◊

Un cas particulier de ce lemme est celui où la restriction d'attribut pour une classe possédant des instances est vide.

**Proposition 3 :**

S'il existe $o \in T_O$, $a \in T_A$, $v \in T_V$ tels que $(o.a=v) \in B$, $(o \in c) \in B$ et $I[v] \notin K_B(a|c)$, alors la base est inconsistante.

*démonstration.* si $I[v] \notin K_B(a|c)$, alors il existe une classe $c'$ et une chaîne de $c$ à $c'$ telles que $(c'.a \subseteq d) \in B$ et $\neg v{:}d$. Par la proposition 1, tout modèle de $B$ doit satisfaire $I[a](I[o]) = I[v] \in I[a|c]$. Or $I[a|c] \subseteq I[d]$ et $\neg v{:}d$. Par conséquent il ne peut exister un tel modèle. ◊

**Proposition 4 :**

S'il existe $o \in T_O$, $a \in T_A$, $v \in T_V$, $v' \in T_V$ tels que $(o.a=v) \in B$ et $(o.a=v') \in B$ avec $v \neq v'$, alors la base est inconsistante.

*démonstration.* Toute fonction d'interprétation $I$ est telle que $I[a]$ est une fonction, ce qui n'est pas possible si $I[a](I[o]) = I[v]$ et $I[a](I[o]) = I[v']$ car $I$ est injective (si $v \neq v'$, alors $I[v] \neq I[v']$). Il ne peut donc pas exister de modèle de $B$. ◊

Il faut noter que la possibilité d'avoir deux valeurs pour le même attribut d'un objet, si elle est exclue par la sémantique (comme le montre cette proposition) ne l'est pas par la syntaxe donnée pour une base (définition 1). Il est donc utile de savoir qu'une base contenant deux assertions de cette forme n'a pas de modèle.

**Proposition 5 :**

Si une base est consistante, les interprétations élémentaires de cette base sont des modèles de la base.

*démonstration.* Pour qu'un couple $\langle D, \hat{I} \rangle$ soit un modèle, il faut et il suffit que $\langle D, \hat{I} \rangle$ soit une interprétation et que la fonction d'interprétation satisfasse toutes les assertions de la base. $\langle D, \hat{I} \rangle$ est une interprétation aux conditions suivantes :

$I_B$: $T_O \rightarrow D$ est une fonction injective. Or, $K_B(o) \in D_K \subseteq D$ donc $I_B$ est injective vu qu'elle n'est définie que sur $D_K$ et qu'elle est bijective sur ce sous-ensemble de $D$.

$\hat{I}$: $T_C \rightarrow 2^D$. Or, pour tout $o$, $K_B(o)$ appartient à $D$, ainsi que tous les éléments notés $o_c$.

$I_B$: $T_A \rightarrow (D \rightarrow \tau)$ : $I_B[a]$ est une fonction aux conditions suivantes :

Pour chaque objet $o$ :

Si $K_B(a)(K_B(o))$ est défini, il n'existe pas $(o.a=v)\in B$ et $(o.a=v')\in B$ avec $v\neq v'$ et de plus $K_B(a)(K_B(o))\in \hat{I}[a|c]$ si $\hat{I}[o]\in \hat{I}[c]$.

sinon $\bigcap_{(o\in c)\in B} \hat{I}[a|c]\neq\emptyset$ pour tout $o$.

Ces conditions sont remplies dans le cas où la base est consistante d'après les propositions 2, 3 et 4.

$<D,\hat{I}>$ est donc une interprétation. Par ailleurs, toute interprétation élémentaire satisfait chacun des types possibles d'assertions si la base est consistante :

$(o\in c)\in B$ : $K_B(o)\in K_B(c)$; par définition de $\hat{I}[c]$, $\hat{I}[o]=K_B(o)\in \hat{I}[c]$ vu que $\hat{I}[c]$ est défini par la troisième alternative de sa définition dans ce cas précis; en effet, $\hat{I}[c]\neq\emptyset$ (d'après la proposition 2 car $(o\in c)\in B$ et que la base est consistante) et $K_B(c)\neq\emptyset$.

$(c\leq c')\in B$ : $K_B(c)\subseteq K_B(c')$. Si $\hat{I}[c]=\emptyset$, $\hat{I}[c]\subseteq\hat{I}[c']$ trivialement; si $\hat{I}[c]\neq\emptyset$, il n'existe pas $a$ tel que $K_B(a|c')=\emptyset$ (car sinon $K_B(a|c)=\emptyset$ et alors $\hat{I}[c]\neq\emptyset$) et il existe $c$ tel que $(c\leq c')\in B$, donc $\hat{I}[c']$ est donné par la troisième alternative de sa définition et donc $\hat{I}[c]\subseteq\hat{I}[c']$.

$(c.a\subseteq d)\in B$ : $K_B(a|c)=\hat{I}[a|c]\subseteq\hat{I}[d]$ par définition de $\hat{I}[a|c]$.

$(o.a=v)\in B$ : $\hat{I}[a](\hat{I}[o])=\hat{I}[v]$, par définition de $\hat{I}[a]$, ce qui est possible car il n'existe pas $(o.a=v)\in B$ et $(o.a=v')\in B$ avec $v\neq v'$, sinon la base est inconsistante (d'après la proposition 4). ◊

La réciproque de cette proposition est immédiate car dès qu'il existe un modèle d'une base, celle-ci est consistante.

### Vocabulaire

Si la base est consistante, toute interprétation élémentaire est appelé modèle élémentaire.

Grâce à la proposition précédente et aux trois propositions qui la précèdent, il est possible de caractériser syntaxiquement l'inconsistance d'une base. Une inconsistance peut apparaître dans une base si un attribut d'instance :

- possède deux valeurs,
- possède une valeur qui est incompatible avec une restriction d'attribut de l'une de ses classes (lorsque la valeur est incompatible avec l'intersection des restrictions pour l'attribut, il existe un domaine particulier auquel elle n'appartient pas),
- ne possède pas de valeur, mais l'ensemble des contraintes portant sur la valeur de l'attribut sont incompatibles. Plus précisément, il existe un objet pour lequel l'intersection des restrictions d'attribut d'un ensemble de classes dont il est

instance directement ou instance de l'une des sous-classes est vide (et donc il n'existe pas de modèle de la base, vu qu'il ne peut y avoir d'interprétation à la valeur d'attribut de l'objet).

Ceci est rassemblé dans la proposition suivante.

**Proposition 6 :**

Une base $B$ est inconsistante si et seulement si l'une des trois conditions suivantes est vérifiée :

- il existe un objet $o$ et un attribut $a$ tels que $(o.a{=}v)\in B$ et $(o.a{=}v')\in B$ avec $v\neq v'$
- il existe un objet $o$, un attribut $a$ et une chaîne d'une classe $c$ à une classe $c'$ (éventuellement confondues) tels que $(o.a{=}v)\in B$, $(o\in c)\in B$, $(c'.a\subseteq d)\in B$ et $\neg v{:}d$
- il existe un objet $o$, un attribut $a$ et un ensemble de chaînes d'une classe d'appartenance de $o$ à des classes $c_1,\ldots, c_n$ tels que $\bigcap_{(c_i.a\subseteq d_{ik})\in B}\hat{I}(d_{ik})=\emptyset$

*démonstration*. La démonstration de la proposition 5 a montré que les seules possibilités d'inconsistance sont :

il existe $o$ et $a$ tels que : $(o.a{=}v)\in B$ et $(o.a{=}v')\in B$ avec $v\neq v'$

ou il existe $o$, $a$ et $c$ tels que : $K_B(a)(K_B(o))\notin\hat{I}[a|c]$ si $\hat{I}[o]\in\hat{I}[c]$

ou il existe $o$ tel que : $\bigcap_{(o\in c)\in B}\hat{I}[a|c]=\emptyset$.

- Le premier cas est trivialement équivalent au premier point de la proposition.
- $\hat{I}[o]\in\hat{I}[c]$ donc il existe une chaîne d'une classe $c'$ à la classe $c$ telle que $(o\in c')\in B$; d'autre part, si $(o.a{=}v)\in B$ $K_B(a)(K_B(o))=I[v]$ et donc $I[v]\notin\hat{I}[a|c]$; vu la définition de $\hat{I}[a|c]$, il existe une chaîne de la classe $c$ à une classe $c''$ telle que $(c''.a\subseteq d)\in B$ et $I[v]\notin I[d]$. En rassemblant ces deux chaînes en une seule :
  il existe une chaîne de classes de $c'$ à $c''$ telle que $(o\in c')\in B$, $(c''.a\subseteq d)\in B$, et $\neg v{:}d$ (car $I[v]\notin I[d]$).
- $\bigcap_{(o\in c)\in B}\hat{I}[a|c]=\bigcap_{(o\in c)\in B}\left(\tau\cap\bigcap_{(c.a\subseteq d_i)\in B}I[d_i]\cap\bigcap_{(c\leq c')\in B}K_B(a|c')\right)=\bigcap_{(o\in c)\in B}\left(\bigcap_{\substack{(c'.a\subseteq d_i)\in B\\ \text{chaîne de } c \text{ à } c'}}I[d_i]\right)$

  Pour que cette intersection soit vide, il suffit qu'un sous-ensemble de cette intersection soit vide, ce qui équivaut au dernier point de la proposition. ◊

Le même type de résultat peut être obtenu pour l'incohérence :

**Proposition 7 :**

Si la base est consistante, les cinq propositions suivantes sont équivalentes :

(1) $c\in T_C$ est une classe incohérente,

(2) $\hat{I}[c]=\emptyset$,

(3) $\exists a$ ; $\hat{I}[a|c]=\emptyset$,

$$(4)\ \ \exists a, \exists c'\ ;\ \begin{cases} \text{il existe une classe } c' \text{ et une chaîne de } c \text{ à } c' \\ \forall c'_1,\ldots,c'_n \text{ tels que } (c' \leq c'_i) \in B,\ \hat{I}[a|c'_i] \neq \emptyset \\ \text{et } \bigcap_{(c'.a \subseteq d_j) \in B} \hat{I}[d_j] \cap \bigcap_{(c' \leq c'_i) \in B} \hat{I}[a|c'_i] = \emptyset \end{cases}$$

(5) $\exists a$, $\exists$ un ensemble de chaînes de $c$ à des classes $c_1,\ldots, c_n$; tels que :

$$\bigcap_{c_i}\ \bigcap_{(c_i.a \subseteq d) \in B} \hat{I}[d] = \emptyset$$

Remarquons que dans le cas d'intervalles d'entiers, cette dernière égalité équivaut au fait qu'il existe deux intervalles particuliers dont l'intersection est vide.

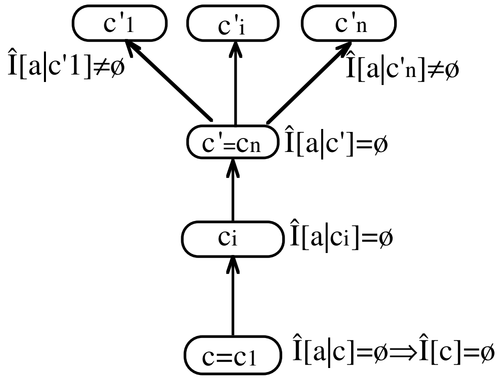


Figure 4 : situation de la proposition 7(4) : pour un ensemble de surclasses de la plus haute surclasse de $c$ dont la restriction de domaine est vide, l'intersection des restrictions de domaine est vide (et chacune des restrictions est non vide)

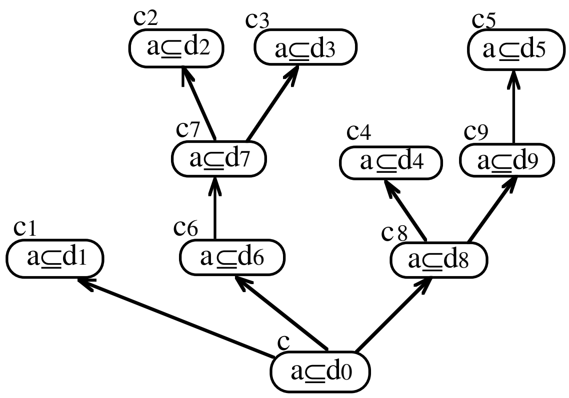


Figure 5 : situation de la proposition 7(5). Il existe une chaîne de la classe $c$ à chacune des classes $c_1$, $c_2$, $c_3$, $c_4$ et $c_5$ et l'intersection de leur restriction de domaine est vide : $\hat{I}(d_1) \cap \hat{I}(d_2) \cap \hat{I}(d_3) \cap \hat{I}(d_4) \cap \hat{I}(d_5) = \emptyset$.

*démonstration.*

(1) $\Rightarrow$ (2) La base étant consistante, $\hat{I}$ est un modèle de la base; $c$ étant incohérente, par définition $\hat{I}[c]=\emptyset$.

(2) $\Rightarrow$ (3) Si $\hat{I}[c]$ est vide, cela signifie qu'il existe un attribut $a$ particulier tel que $\hat{I}[a|c]$ est vide : $\hat{I}[a|c] = \emptyset$.

(3) ⇒ (1) Par la proposition 1, $I[c] \subseteq \{i \in D,\ I[a](i) \in I[a|c]\}$ et par (3) $I[a|c]=\varnothing$. Par conséquent, il ne peut pas exister de tels $i \in D$ et donc $I[c]=\varnothing$ pour tout modèle. C'est donc une classe incohérente.

(3) ⇒ (4) S'il existe un attribut $a$ particulier tel que $K_B(a|c)$ soit vide.

$K_B(a|c) = \tau \cap \bigcap_{(c.a \subseteq d_i) \in B} I[d_i] \cap \bigcap_{(c \leq c') \in B} K_B[a|c'] = \varnothing$ et donc soit,

a) $\exists c'$ ; $(c \leq c') \in B$ et $\hat{I}[a|c_i] = \varnothing$ et le raisonnement peut être repris à partir de $c'$;

b) $\forall c_1, \ldots, c_n$; $(c \leq c_i) \in B$, $\hat{I}[a|c_i] \neq \varnothing$ et la propriété est démontrée.

Puisque le graphe de $\leq$ est sans cycle et que $T_C$ est fini, cette récurrence est bien fondée et démontre la propriété.

(4) ⇒ (3) $\hat{I}[a|c'] = \varnothing$ donc en appliquant la définition de $I[a|c]$ sur les $c_1, \ldots, c_n$ tels que $c_1=c$, $c_n=c'$ et $\forall i \in [1..n\text{-}1]$ $(c_i \leq c_{i+1}) \in B$, $\hat{I}[a|c_i] = \varnothing$. Or $c=c_1$ par conséquent $\hat{I}[a|c]=\varnothing$.

(3) ⇒ (5) S'il existe un attribut $a$ particulier tel que $K_B(a|c)$ est vide, cela signifie que :

$$K_B(a|c) = \tau \cap \bigcap_{(c.a \subseteq d_i) \in B} \hat{I}[d_i] \cap \bigcap_{(c \leq c') \in B} K_B[a|c']$$

$$= \tau \cap \bigcap_{(c.a \subseteq d_i) \in B} \hat{I}[d_i] \cap \bigcap_{c' \text{ t.q. } \exists \text{ une chaîne de } c \text{ à } c'} \ \bigcap_{(c'.a \subseteq d) \in B} \hat{I}[d] = \varnothing,$$

ce qui correspond à l'égalité du point (5).

(5) ⇒ (3) Immédiat en utilisant les équations de la preuve précédente. ◊

La proposition peut être reformulée de la manière suivante : si (1) $c$ est une classe incohérente, alors (2) son interprétation élémentaire est vide, (3) il existe un attribut pour lequel l'interprétation est vide, (4) il existe une classe incohérente pour laquelle les surclasses sont cohérentes et (5) il existe un ensemble de contraintes pour cet attribut dans les surclasses de $c$ qui sont incompatibles entre elles.

## 1.4. Déduction syntaxique

Calculer tous les modèles d'une base de connaissance pour en connaître les conséquences est très coûteux. Afin d'accéder à ces conséquences, un système déductif est défini par un ensemble de règles d'inférence. La complétude sera étudiée au paragraphe suivant (§1.5).

$\dfrac{\delta_1 \ldots \delta_n}{\delta}\ co$ signifie que si $\delta_1, \ldots \delta_n$ sont satisfaites et que la condition est vérifiée alors $\delta$ est également satisfaite. La condition $co$ est distinguée des assertions car les conditions (sur les types) ne sont pas des assertions de la base.

**Définition 16 : règles d'inférence**

Pour tout $c$, $c'$, $c''$, $c_i \in T_C$, $o \in T_O$, $a \in T_A$, $v \in T_V$ et $d$, $d'$, $d_i \in T_T$ :

(1) ($\leq$-réflexivité) $$\frac{}{c \leq c}$$

(2) ($\leq$-transitivité) $$\frac{c \leq c' \quad c' \leq c''}{c \leq c''}$$

(3) ($\in$-clôture) $$\frac{c \leq c' \quad o \in c}{o \in c'}$$

(4) (d-intersection) $$\frac{c.a \subseteq d \quad c.a \subseteq d'}{c.a \subseteq d \wedge_\tau d'}$$

(5) (d-héritage) $$\frac{c.a \subseteq d \quad c' \leq c}{c'.a \subseteq d}$$

(6) (d-expansion) $$\frac{c.a \subseteq d}{c.a \subseteq d'} \quad d <:_\tau d'$$

(7) (o-valuation) $$\frac{\{c_i.a \subseteq d_i \quad o \in c_i\}}{o.a = v} \quad \{v\} = \bigwedge_i d_i$$

Les deux premières règles expriment la réflexivité et la transitivité de la relation de spécialisation, la troisième qu'un objet est instance de toutes les surclasses de la classe à laquelle il est attaché. Les quatrième et cinquième règles signifient que la contrainte portant sur un attribut d'une classe est l'accumulation de toutes celles posées sur cet attribut dans la classe et dans ses surclasses. La sixième règle est une règle technique permettant d'obtenir la complétude (elle permet d'ajouter des contraintes plus faibles que la contrainte existante). Enfin, la dernière règle indique que si, pour un ensemble de classes, l'intersection des restrictions de domaine d'un attribut est réduit à un singleton, les objets qui appartiennent à l'ensemble de ces classes possèdent cette valeur pour l'attribut considéré.

La clôture déductive d'une base est l'ensemble unique obtenu en appliquant les règles tant que cela est possible.

**Définition 17 : clôture d'une base**

Soit $B$ une base, la clôture de la base $B$ est la base, notée Cn($B$), telle que :

(1) $B \subseteq \mathrm{Cn}(B)$,

(2) Pour tous $\delta_1, \ldots \delta_n \in \mathrm{Cn}(B)$ et toute règle $\frac{\delta_1 \ldots \delta_n}{\delta} co$, si $co$ est satisfaite, $\delta \in \mathrm{Cn}(B)$,

(3) Cn($B$) ne contient pas d'autres assertions.

**Définition 18 : déduction**

Une base $B$ permet de déduire une assertion $\delta$, noté $B \vdash \delta$, ssi $\delta \in \mathrm{Cn}(B)$.

Une base $B$ permet de déduire une base $B'$, noté $B \vdash B'$, ssi pour tout $\delta \in B'$, $B \vdash \delta$.

La négation de la déduction est notée $\nvdash$.

La déduction permet d'inférer les conséquences de la base. Dans les systèmes à objets, elle correspond à l'héritage, à l'inférence de type ou de valeur. Remarquons que toute assertion appartenant à $\mathrm{Cn}(B)$ est déductible en utilisant un nombre fini de pas de déduction.

Les propositions suivantes établissent les propriétés de système déductif de la déduction.

**Proposition 8 :**

$\vdash$ (considérée comme relation entre bases) est transitive.

*démonstration*. Soient $B$, $B'$ et $B''$ telles que $B \vdash B'$ et $B' \vdash B''$. $\forall \delta \in B''$; $B' \vdash \delta$, alors soit $\delta \in B'$ (et par conséquent $B \vdash \delta$); soit $\delta \notin B'$ et est donc déduite de $B'$ par application des règles ci-dessus à partir d'un ensemble d'assertions appartenant à $B'$. Or, cet ensemble d'assertions de $B'$ peut être déduit par application des règles à partir d'un ensemble d'assertions appartenant à $B$. Par conséquent, l'application des règles permet de déduire $\delta$ à partir de $B$ et donc $B \vdash B''$. ◊

**Proposition 9 :**

Soient $B$ et $B'$ deux bases, $B \vdash B'$ ssi $B \vdash \mathrm{Cn}(B')$.

*démonstration.*

si) $B' \subseteq \mathrm{Cn}(B')$.

seulement si) $B' \vdash \mathrm{Cn}(B')$ et $\vdash$ est transitive. ◊

## 1.5. Complétude

Établir le pont entre la déduction syntaxique ($\vdash$) et la conséquence sémantique ($\models$) permet de s'assurer que ce qui est déduit syntaxiquement est bien valide sémantiquement et vice versa. En particulier, une caractérisation syntaxique d'une base inconsistante ou incohérente permet de ne pas utiliser la notion de modèle pour détecter l'inconsistance ou l'incohérence.

La correction est assez naturellement acquise, vu que les règles d'inférence décrivent un mécanisme de déduction conforme aux propriétés classiques d'une fonction d'interprétation.

**Proposition 10 : correction**

Soient $B$ une base et $\delta$ une assertion, si $B \vdash \delta$ alors $B \models \delta$.

*démonstration*. Soit $\delta$ tel que $B \vdash \delta$. $B \models \delta$ peut être prouvé par récurrence sur l'arbre de déduction. Cet arbre a une taille finie car toute déduction se fait en un nombre fini d'application de règles. En effet, les classes et objets d'une base sont en nombre fini et dans les règles, l'utilisation des assertions sur les domaines les plus restreints pour chaque couple classe-attribut est suffisante, donc un nombre fini d'assertions est nécessaire pour calculer chaque élément de Cn($B$). La récurrence se fait sur la hauteur ($p$) de l'arbre développé pour déduire $\delta$.

$p$=0 : $\delta \in B$ donc $B \models \delta$.

hypothèses de récurrence : pour tout $f$ telle que $B \vdash f$ en utilisant un arbre de hauteur $p$, $B \models f$.

Soit $\delta$ tel que $B \vdash \delta$ en utilisant un arbre de déduction de hauteur $p+1$; la dernière règle de déduction utilisée est de la forme : $\dfrac{\delta_1 \ldots \delta_n}{\delta}$, avec $\delta_1$, $\ldots\delta_n$ déduits à partir d'un arbre de hauteur au plus $p$ (et dont l'éventuelle condition est vraie). Par hypothèse de récurrence $B \models \delta_1$, … $B \models \delta_n$. Il faut montrer que $B \models \delta$. La proposition se démontre pour chacune des règles d'inférence :

- $\leq$-réflexivité : si $B \vdash c \leq c$, comme $I[c] \subseteq I[c]$, $B \models c \leq c$.
- $\leq$-transitivité : si $B \vdash c \leq c''$ avec $B \models c \leq c'$, $B \models c' \leq c''$, alors $\forall M$=<$D$,$I$> tel que $\models_M B$, $I[c] \subseteq I[c']$ et $I[c'] \subseteq I[c'']$, d'où $I[c] \subseteq I[c'']$ et donc $\forall$ $M$ modèle de $B$, $\models_M c \leq c''$ d'où : $B \models c \leq c''$.
- $\in$-clôture : si $B \vdash o \in c'$ avec $B \models o \in c$ et $B \models c \leq c'$, alors $\forall M$=<$D$,$I$> tel que $\models_M B$, $I[c] \subseteq I[c']$ et $I[o] \in I[c]$, d'où $I[o] \in I[c']$; donc, $\forall M$ modèle de $B$, $\models_M o \in c'$, et par conséquent $B \models o \in c'$.
- d-intersection : si $B \vdash c.a \subseteq d \wedge d'$ avec $B \models c.a \subseteq d$ et $B \models c.a \subseteq d'$ alors $\forall M$=<$D$,$I$> tel que $\models_M B$, $I[a|c] \subseteq I[d]$ et $I[a|c] \subseteq I[d']$, d'où $I[a|c] \subseteq I[d] \cap I[d'] = I[d \wedge d']$. Donc $\forall M$=<$D$,$I$>, $\models_M c.a \subseteq d \wedge d'$ et donc $B \models c.a \subseteq d \wedge d'$.
- d-héritage : si $B \vdash c'.a \subseteq d$ avec $B \models c.a \subseteq d$ et $B \models c' \leq c$, alors $\forall M$=<$D$,$I$> tel que $\models_M B$, $I[a|c] \subseteq I[d]$ et $I[a|c'] \subseteq I[a|c]$, d'où $I[a|c'] \subseteq I[d]$ $\forall M$ modèle de $B$, $\models_M c'.a \subseteq d$ donc $B \models c'.a \subseteq d$.
- d-expansion : si $B \vdash c.a \subseteq d'$ et $d$<:$d'$ avec $B \models c.a \subseteq d$, alors $\forall M$=<$D$,$I$> tel que $\models_M B$, $I[a|c] \subseteq I[d]$ et $I[d] \subseteq I[d']$, d'où $I[a|c] \subseteq I[d']$. Par conséquent, $\forall M$ modèle de $B$, $\models_M c.a \subseteq d'$ et donc $B \models c.a \subseteq d'$.

o-valuation : si $B\vdash o.a{=}v$ avec pour $i\in[1..n]$ $B\models c_i.a\subseteq d_i$, $B\models o\in c_i$ et $\bigwedge_i d_i=\{v\}$ alors $\forall M{=}<D,I>$ tel que $\models_M B$, $I[a|c_i]\subseteq I[d_i]$ d'où $I[a](I[o])\in I[d_i]$ pour $i\in[1..n]$; donc $I[a](I[o])\in I[\bigwedge_i d]=I[v]$. Ce qui signifie que $\forall M$ modèle de $B$, $\models_M o.a{=}v$, donc $B\models o.a{=}v$.

Conclusion de la récurrence : $\forall\ \delta$ et $p$ tels que $B\vdash\delta$ en utilisant un arbre de déduction de hauteur $p$ (pour toutes les assertions, il existe un tel arbre de déduction), alors $B\models\delta$. Par conséquent, si $B\vdash\delta$ alors $B\models\delta$. ◊

L'expression syntaxique de l'inconsistance est plus délicate que dans les systèmes logiques car elle se traduit par trois cas distincts déjà évoqués dans la proposition 6.

**Proposition 11 : inconsistance syntaxique**

$B$ est inconsistante ssi

il existe $o$, $a$, $v$ et $v'$ tels que $B\vdash o.a{=}v$ et $B\vdash o.a{=}v'$ et $v\neq v'$
ou il existe $o$, $c$, $a$, $d$ et $v$ tels que $B\vdash o\in c$, $B\vdash c.a\subseteq d$, $B\vdash o.a{=}v$, et $\neg v{:}d$
ou il existe $o$, $\{c_i\}$, $a$ tels que $B\vdash o\in c_i$, $B\vdash c_i.a\subseteq d_i$ et $\bigwedge_i d_i=\perp_\tau$

*démonstration.*

si) Dans chacun des trois cas, il est impossible de trouver un modèle :

1) Si $B\vdash o.a{=}v$ et $B\vdash o.a{=}v'$, alors $\hat{I}[a](\hat{I}[o])=\hat{I}[v]$, $\hat{I}[a](\hat{I}[o])=\hat{I}[v']$ et $v\neq v'$ ce qui est contradictoire avec le fait que $\hat{I}$ soit injective. Donc, il ne peut exister de modèle $\hat{I}$.

2) Si $B\vdash o.a{=}v$, $B\vdash c.a\subseteq d$, $B\vdash o\in c$ et $\neg v{:}d$, alors, si la base est consistante, $B\models o.a{=}v$, $B\models c.a\subseteq d$, $B\models o\in c$ et donc $\hat{I}[v]=\hat{I}[a](\hat{I}[o])\in\bigcap_i\hat{I}[d_i]=\emptyset$, ce qui n'est pas possible, donc la base est inconsistante.

3) Si $B\vdash o\in c_i$ et $B\vdash c_i.a\subseteq d_i$, alors $\hat{I}[o]\in\hat{I}[c_i]$, $\hat{I}[a](\hat{I}[o])\in\bigcap_i\hat{I}[a|c_i]=\emptyset$, donc $\hat{I}$ n'est pas un modèle.

seulement si) Par la proposition 6, il existe exactement trois cas d'inconsistance :

1) $(o.a{=}v)\in B$, $(o.a{=}v')\in B$ et $v\neq v'$. Dans ce cas $B\vdash o.a{=}v$ et $B\vdash o.a{=}v'$.

2) $(o.a{=}v)\in B$, $(o\in c)\in B$ et il existe une chaîne de la classe $c$ à une classe $c'$ (éventuellement confondues) $(c'.a\subseteq d)\in B$ et $\neg v{:}d$. Dans ce cas, par l'application répétée de la règle de ($\leq$-transitivité) sur les couples $(c_i, c_{i+1})$ de la chaîne de classes, $B\vdash c\leq c'$ et par l'application de la ($\in$-clôture), $B\vdash o\in c'$.

3) il existe $a$ et un ensemble de chaînes d'une classe d'appartenance de $o$ à des classes $c_1,\ldots, c_n$ tels que $\bigcap_{(c_i.a\subseteq d_k)\in B}\hat{I}(d_k)=\emptyset$.

L'application répétée de la règle de ($\in$-clôture) donne $B\vdash o\in c_i$, $B\vdash c_i.a\subseteq di$ et $\bigwedge_i d_i=\perp_\tau$ (à noter que l'ensemble $\{c_i\}$ n'est pas unique).

Ces cas se ramènent donc aux expressions données dans la proposition. ◊

Cette proposition se vérifie sur l'exemple de base inconsistante (figure 2). En effet :
$(i''.a=7)\in B$, donc $B\vdash i''.a=7$,
$(i''\in c_4)\in B$ et $(c_4\leq c_2)\in B$ donc, d'après la règle de ($\in$-clôture), $B\vdash i''\in c_2$,
$(c_2.a\subseteq[10\ 25])\in B$ donc $B\vdash c_2.a\subseteq[10\ 25]$ et $\neg 7{:}[10\ 25]$.

La complétude faible est l'inverse de la correction (tout ce qui est vrai dans tous les modèles est dérivable). Elle est appelé ainsi pour la distinguer de la complétude forte qui rassemble correction et complétude faible. La complétude faible n'est acquise que lorsque la base est consistante. En effet, une base inconsistante n'ayant pas de modèle, n'importe quelle assertion est sa conséquence, ce qui n'est pas le cas des règles d'inférence. Ceci est une propriété de localité typique des systèmes à objets. La proposition précédente est alors indispensable car elle permet de caractériser syntaxiquement l'inconsistance.
Les lemmes suivants seront utilisés pour démontrer la complétude faible.

**Lemme 12 :**
Si la base $B$ est consistante, si $B\models c\leq c'$ et si $c\neq c'$ alors il existe une chaîne de $c$ à $c'$.

*démonstration.* Par l'absurde : s'il n'existe pas une telle chaîne (définition 13), alors il existe un modèle de $B$ tel que $I[c]\not\subset I[c']$ (comme la base est consistante, $<D,\hat{I}>$ est un modèle de la base). Une interprétation $M=<D',I'>$, basée sur le modèle élémentaire $<D,\hat{I}>$, est définie par :

- $D'$ est augmenté de $i$ tel que $i\notin D$,
- pour toutes les classes telles qu'il existe une chaîne de $c$ à $c''$, $I'[c'']=\{I'[o] \text{ tel que } (o\in c'')\in B\}\cup\bigcup_{(c'''\leq c'')\in B} I'[c''']\cup\{i\}$,
- les autres définitions n'étant pas modifiées.

Donc $\forall c''$ telle qu'il existe une chaîne de $c$ à $c''$, $i\in I'[c'']$et $i\notin I'[c']$ car il n'existe pas de chaîne de $c$ à $c'$. Donc $I'[c]\not\subset I'[c']$ et $M=<D',I'>$ est un modèle de $B$ (car la preuve des différentes assertions n'est pas modifiée par ces changements de définitions par rapport au modèle élémentaire), donc $B\not\models c\leq c'$ d'où l'absurdité. ◊

**Lemme 13 :**
Si la base $B$ est consistante et si $B\models o\in c$ alors il existe $c'$ tel que $B\models c'\leq c$ et $(o\in c')\in B$.

*démonstration.* Par l'absurde : comme la base est consistante, $<D,\hat{I}>$ est un modèle de la base. $\hat{I}[o]\in\hat{I}[c]$ et $\hat{I}[c]=\{\hat{I}[o'] \text{ tel que } (o'\in c)\in B\}\cup\bigcup_{(c'\leq c)\in B}\hat{I}[c']$; en effet si $\hat{I}[c]=\{o_c\}$ ou

$\hat{I}[c]=\emptyset$, alors $B\not\models o\in c$. Donc $\hat{I}[o]\in\{\hat{I}[o'] \text{ tel que } (o'\in c)\in B\}$ ou $\hat{I}[o]\in\hat{I}[c']$ avec $(c'\leq c)\in B$. Dans le premier cas, le résultat est prouvé; dans le second cas, le résultat se montre par récurrence le long d'une chaîne de $c$ à $c'$ telle que $\hat{I}[o]\in\hat{I}[c']$ (il existe une telle classe sinon $\hat{I}[o]\notin\hat{I}[c]$). On a alors $(o\in c')\in B$ et $B\models c'\leq c$ car il existe une chaîne de $c$ à $c'$. ◊

**Lemme 14 :**

Si la base $B$ est consistante et si $B\models c.a\subseteq d$; alors

soit $d=\bigwedge_{(c.a\subseteq d_i)\in B} d_i \wedge \bigwedge_{\substack{B\models c\leq c_j\\(c_j.a\subseteq d_{jk})\in B}} d_{jk}$

soit $d$ est un domaine non minimal : $\exists d'<:_\tau d\ B\models c.a\subseteq d'$

*démonstration*. Si $d$ est minimal, c'est-à-dire si $\forall d'<:_\tau d\ B\not\models c.a\subseteq d'$, montrons alors que $d=\bigwedge_{(c.a\subseteq d_i)\in B} d_i \wedge \bigwedge_{\substack{B\models c\leq c_j\\(c_j.a\subseteq d_{jk})\in B}} d_{jk}=\Delta$.

- $B\models c.a\subseteq\Delta$ (car $B\models c.a\subseteq d_i$ pour tout $(c.a\subseteq d_i)\in B$ et $B\models c.a\subseteq d_{\mathrm{jk}}$ pour tout $B\models c\leq c_j$ et $(c_j.a\subseteq d_{\mathrm{jk}})\in B$) donc $\neg\Delta<:_\tau d$.
- $\Delta$ est minimal car

$$\begin{aligned}\hat{I}[a|c] &= \tau\cap\bigcap_{(c.a\subseteq d_i)\in B}\hat{I}[d_i]\cap\bigcap_{(c\leq c')\in B}\hat{I}[a|c']\\ &=\tau\cap\bigcap_{(c.a\subseteq d_i)\in B}\hat{I}[d_i]\cap\bigcap_{\substack{B\models c\leq c'\\(c_j.a\subseteq d_{jk})\in B}}\hat{I}[d_{jk}]=\hat{I}[\Delta],\end{aligned}$$

  $\hat{I}[a|c]\not\subset\hat{I}[d']$ et $<D,\hat{I}>$ est modèle (car la base est consistante) donc $\not\models_{\hat{I}} c.a\subseteq d'$ pour $d'<:_\tau\Delta$.
- Or $\models_{\hat{I}} c.a\subseteq\Delta\wedge_\tau d$, donc $\neg\Delta\wedge_\tau d<:_\tau\Delta$ (d'après le second point), d'où $\Delta\wedge_\tau d=_\tau d$. De plus, $\neg\Delta<:_\tau d$ (d'après le premier point) donc $d=\Delta$. ◊

**Lemme 15 :**

Si la base $B$ est consistante, $B\models o.a=v$ et $(o.a=v)\notin B$ alors il existe $c_1,\ldots,c_n$ tels que $\forall i$ $B\models o\in c_i$, $B\models c_i.a\subseteq d_i$ et $\bigwedge_i\{d_i\}=\{v\}$ (avec $(\wedge d_i)\text{-}d_j\neq\{v\}$).

*démonstration*. Par l'absurde : Soit $B\models o.a=v$, $(o.a=v)\notin B$ et $\forall c_i$, $B\models o\in c_i$ et $B\models c_i.a\subseteq d_i$ et $\bigwedge_i\{d_i\}\neq\{v\}$. Il est alors possible de construire un modèle élémentaire tel que $o.a\in\bigwedge_i\{d_i\}-\{v\}$; le modèle élémentaire étant un modèle (car la base est consistante), cela a pour conséquence que $B\not\models o.a=v$ ce qui est contraire à l'hypothèse. ◊

**Proposition 16 : complétude faible**

Soient une base $B$ consistante et une assertion $\delta$, si $B\models\delta$ alors $B\vdash\delta$.

*démonstration de la complétude faible*. Ce théorème est démontré en 4 étapes qui correspondent à la démonstration pour chacun des 4 types d'assertions possibles d'une base. L'ordre de démonstration est basé sur la remarque suivante : les assertions du type $c \leq c'$ ne peuvent être déduites d'aucune autre; les assertions du type $o \in c$ et du type $c.a \sqsubseteq d$ peuvent être déduites en utilisant des assertions du type $c \leq c'$; les assertions du type $o.a = v$ peuvent être déduites en utilisant des assertions du type $o \in c$ et du type $c.a \sqsubseteq d$. Ainsi, lorsque l'un des cas est démontré, il est utilisable dans la suite de la démonstration. De plus, comme la base est consistante, $\langle D, \hat{I} \rangle$ est un modèle de la base.

$B \models c \leq c' \Rightarrow B \vdash c \leq c'$ : Si $B \models c \leq c$, alors le résultat est acquis par la règle de ($\leq$-reflexivité). D'après le lemme 12, si $B \models c \leq c'$, et $c \neq c'$ alors il existe une chaîne de $c$ à $c'$. En appliquant la règle de ($\leq$-transitivité) sur la chaîne, $B \vdash c \leq c'$ est obtenu.

$B \models o \in c \Rightarrow B \vdash o \in c$ : les conditions du lemme 13 sont vérifiées, ainsi il existe $c'$ tel que $B \models c' \leq c$ et $(o \in c') \in B$. D'où $B \vdash c' \leq c$ (d'après le point précédent de la démonstration) et $B \vdash o \in c'$. En appliquant la règle de ($\in$-clôture), $B \vdash o \in c$.

$B \models c.a \sqsubseteq d \Rightarrow B \vdash c.a \sqsubseteq d$ : soit $d$ tel que $B \models c.a \sqsubseteq d$; d'après le lemme 14, $d = \bigwedge_{(c.a \sqsubseteq d_i) \in B} d_i \wedge \bigwedge_{\substack{B \models c \leq c_j \\ (c_j.a \sqsubseteq d_{jk}) \in B}} d_{jk}$ (1) ou $d$ est non minimal (2).

(1) si $d$ est minimal, $B \models c.a \sqsubseteq \bigwedge_{(c.a \sqsubseteq d_i) \in B} d_i \wedge \bigwedge_{\substack{B \models c \leq c_j \\ (c_j.a \sqsubseteq d_{jk}) \in B}} d_{jk}$; pour tout $B \models c \leq c_j$ et $(c_j.a \sqsubseteq d_{jk}) \in B$, $B \vdash c \leq c_j$ (démontré ci-dessus) et $B \vdash c_j.a \sqsubseteq d_{jk}$ et donc en utilisant la règle de (d-héritage), $B \vdash c.a \sqsubseteq d_{jk}$. De plus, pour tout $(c.a \sqsubseteq d_i) \in B$, $B \vdash c.a \sqsubseteq d_i$ et donc en utilisant la règle de (d-intersection) sur toutes les assertions du type $c.a \sqsubseteq d$ pour $c$ et $a$ donnés et $d$ quelconque :

$$B \vdash c.a \sqsubseteq \bigwedge_{(c.a \sqsubseteq d_i) \in B} d_i \wedge \bigwedge_{\substack{B \models c \leq c_j \\ (c_j.a \sqsubseteq d_{jk}) \in B}} d_{jk},$$

d'où $B \vdash c.a \sqsubseteq d$

(2) Si $d$ est tel que $B \models c.a \sqsubseteq d$ et $\exists d' <:_\tau d$ tel que $B \models c.a \sqsubseteq d'$, il existe (au moins) un $d'' <:_\tau d'$ «minimal» (c'est-à-dire tel que $B \models c.a \sqsubseteq d''$ et $\forall d' <:_\tau d''$ $B \not\models c.a \sqsubseteq d'$) et donc $B \vdash c.a \sqsubseteq d''$ d'après ce qui précède et par la règle de (d-expansion) $B \vdash c.a \sqsubseteq d$.

$B \models o.a = v \Rightarrow B \vdash o.a = v$ : Si $(o.a = v) \in B$ le résultat est immédiat. Si $(o.a = v) \notin B$ alors les prémisses du lemme 15 sont satisfaites, et donc il existe $c_1, \ldots, c_n$ tels que $\forall i$ $B \models o \in c_i$ et $B \models c_i.a \sqsubseteq d_i$ et $\bigwedge_i \{d_i\} = \{v\}$. Par conséquent, $\forall i$ $B \vdash o \in c_i$ et $B \vdash c_i.a \sqsubseteq d_i$, d'après ce qui a été démontré précédemment et en appliquant la règle de (o-valuation), $B \vdash o.a = v$.

Conclusion : pour chacun des quatre types d'assertion, le théorème est démontré. ◊

De même qu'il existe une notion d'inconsistance syntaxique, il est intéressant de caractériser syntaxiquement l'incohérence pour la détecter et l'éviter.

**Proposition 17 : incohérence syntaxique**

$B$ est incohérente ssi il existe $c \in T_C$ et $a \in T_A$ tels que $B \vdash c.a \subseteq \perp$.

*démonstration.*

si) Si $B \vdash c.a \subseteq \perp$, alors $B \models c.a \subseteq \perp$, d'après la proposition 10. Donc, pour tous les modèles, $I[a|c]=\emptyset$ et par conséquent $I[c]=\emptyset$.

seulement si) $\exists c$; pour tous les modèles, $I[c]=\emptyset$. Par la proposition 7,

$$\exists a;\ \hat{I}[a|c] = \tau \cap \bigcap_{(c.a\subseteq d_i)\in B} \hat{I}[d_i] \cap \bigcap_{(c\leq c')\in B} \hat{I}[a|c'] = \emptyset$$

ce qui signifie que :

$$[\bigwedge_{(c.a\subseteq d_i)\in B} d_i] \wedge [\bigwedge_{\substack{(c\leq c')\in B\\ B\vdash c'.a\subseteq d'_i}} d'_i] = \perp.$$

Or, $B \vdash c.a \subseteq \bigwedge_{(c.a\subseteq d_i)\in B} d_i$ par la règle de (d-intersection), $B \vdash c.a \subseteq \bigwedge_{\substack{(c\leq c')\in B\\ B\vdash c'.a\subseteq d'_i}} d'_i$ par la règle de (d-héritage) et, par conséquent, $B \vdash c.a \subseteq \perp$ de nouveau par la règle de (d-intersection). ◊

## 1.6. Forme première

Les systèmes de représentation de connaissance ne manipulent pas les bases sous forme de modèle ou de clôture déductive (souvent infinie dans le cas simple présenté ici à cause des intervalles d'entiers), mais sous une forme normalisée permettant de déterminer rapidement inconsistance et conséquence. La forme première, qui sera utilisée lors de la révision, permet de ramener une base de connaissance à une forme unique de taille restreinte, qui contient suffisamment d'assertions pour déduire la base. Elle supprime de la clôture déductive l'ensemble des assertions qui sont déductibles par les règles d'inférence.

**Définition 19 : forme première**

Soient une base $B$ et $\mathrm{Cn}'(B)=\mathrm{Cn}(B)-\{c\leq c \in \mathrm{Cn}(B)\}$, la réduction réflexive de $\mathrm{Cn}(B)$.

La forme première d'une base $B$, notée $\mathrm{FP}(B)$, se définit par :

$\mathrm{FP}(B) = \mathrm{Cn}'(B)$

$- \{(c \leq c''); (c \leq c'), (c' \leq c'') \in \mathrm{Cn}'(B)\}$ (≤-réduction)

$- \{(o \in c); (o \in c'), (c' \leq c) \in \mathrm{Cn}'(B)\}$ (∈-réduction)

$- \{(c.a \subseteq d); (c.a \subseteq d') \in \mathrm{Cn}'(B)$ et $d' <:_\tau d\}$ (d-normalisation)

– $\{(c.a \subseteq d); \{c_i.a \subseteq d_i \quad c \leq c_i\} \in \mathrm{Cn}'(B), \ \bigwedge_i d_i = d\}$ (d-anti-intersection)

– $\{(o.a = v); \{c_i.a \subseteq d_i \quad o \in c_i\} \in \mathrm{Cn}'(B)$ et $\{v\} = \bigwedge_i d_i\}$ (o-anti-valuation)

Il est nécessaire de considérer l'ensemble Cn($B$), car la réduction réflexive doit être réalisée avant les autres pour ne faire que les retraits pertinents (sinon toutes les assertions de la forme ($c \leq c'$), par exemple, seraient supprimées par la règle de (≤-réduction)).

Les retraits sont effectués pour ne garder dans la forme première que ce qui est strictement nécessaire pour pouvoir redéduire la base entière, c'est-à-dire les assertions qui indiquent l'attachement d'une instance ou d'une classe à la plus basse classe possible, la valeur d'un attribut d'une instance si celle-ci n'est pas déductible des classes auxquelles appartient l'instance, la restriction de domaine d'un attribut de classe si celle-ci n'est pas présente dans l'une de ses surclasses ou calculable par intersection des domaines des surclasses (ce qui se distingue du cas précédent dans le cas de la multi-spécialisation). Les règles de (d-intersection) et de (d-héritage) ne trouvent pas leur pendant direct dans cette définition mais sont regroupées dans la règle de (d-anti-intersection). Il s'agit en effet de ne pas supprimer trop d'assertions sur les restrictions d'attribut (comme cela est montré dans l'exemple qui suit)

La forme calculée comme précédemment mais sans appliquer la (d-anti-intersection) est la *forme normale* de la base, c'est-à-dire celle dans laquelle il existe une et une seule assertion de restriction par attribut et par classe (les autres assertions étant les mêmes que celles de la forme première).

Les exemples qui suivent permettent de se rendre compte du bien-fondé de cette définition de forme première.

**Exemple : la règle de (d-anti-intersection)**

Soit la base : $B$={$c$≤$c'$, $c'.a$⊆[0 20], $c.a$⊆[10 30]}

Cn($B$)={$c$≤$c'$, $c'.a$⊆[0 20], $c.a$⊆[10 30], $c.a$⊆[0 20], $c.a$⊆[10 20]}

En appliquant la règle de (d-anti-intersection), FP($B$)={$c$≤$c'$, $c'.a$⊆[0 20]} et toutes les informations sur la restriction de l'attribut $c$ sont perdues.

**Exemple :**

La base prise comme exemple jusqu'à présent :

$B$={$c_1 \leq c_0$ ; $c_2 \leq c_1$ ; $c_3 \leq c_2$ ; $c_4 \leq c_2$ ; $i \in c_2$ ; $i' \in c_3$ ; $i'' \in c_4$ ; $c_0.a$⊆[-∞ +∞] ; $c_1.a$⊆[0 100] ; $c_2.a$⊆[10 25] ; $c_4.a$⊆[5 15] ; $i.a$=24 ; $i'.a$=18}

est déjà pratiquement sous forme première (ce qui montre l'aspect intuitif de la forme première) : FP($B$)={$c_1 \leq c_0$ ; $c_2 \leq c_1$ ; $c_3 \leq c_2$ ; $c_4 \leq c_2$ ; $i \in c_2$ ; $i' \in c_3$ ; $i'' \in c_4$ ; $c_0.a$⊆[-∞ +∞] ; $c_1.a$⊆[0 100] ; $c_2.a$⊆[10 25] ; **$c_4.a$⊆[10 15]** ; $i.a$=24 ; $i'.a$=18}.

**Proposition 18 :**

Soit une base $B$, FP($B$) est unique.

*démonstration*. Le résultat est trivial vu que la forme première est définie comme un retrait complètement déterministe d'un ensemble d'assertions de Cn(B). ◊

**Proposition 19 :**

La taille maximale de la forme première est de l'ordre de :

$$\begin{cases} (|T_O|+|T_C|)*(|T_A|+|T_C|) \text{ dans le cas général} \\ (|T_O|*|T_A|)+|T_C|*(|T_A|+|T_C|) \text{ sans multi-instanciation} \\ |T_O|*(|T_A|+|T_C|)+(|T_C|*|T_A|) \text{ sans multi-spécialisation} \\ (|T_O|+|T_C|)*|T_A| \text{ sans aucun des deux.} \end{cases}$$

*démonstration*. La forme première ne préserve plus qu'une valeur par attribut d'objet ($|T_O|*|T_A|$), au maximum un domaine par attribut de classe ($|T_C|*|T_A|$) (en effet, la clôture contient toutes les intersections possibles de domaines pour *c.a* donné, donc en particulier l'intersection de toutes les restrictions de domaine, qui est contenu dans tous les autres domaines). Par ailleurs, toute classe peut être attachée à toute autre ($|T_C|^2$, même si l'on considère qu'il ne reste que la réduction transitive de l'ordre); cependant, si l'on se restreint à la mono-spécialisation, cette valeur tombe à $|T_C|$. Tout objet peut être attaché à toute classe ($|T_O|*|T_C|$), mais en se restreignant à la mono-instanciation, la valeur devient $|T_O|$. ◊

L'intérêt théorique principal de la forme première réside dans sa minimalité qui fait que les retraits y sont effectifs (c'est-à-dire que, si une assertion est supprimée d'une base sous forme première, celle-ci ne peut plus être déduite).

**Proposition 20 : minimalité de la forme première**

Pour tout assertion $\delta$ appartenant à FP($B$), FP($B$)–{$\delta$} $\nvdash$ $\delta$.

*démonstration*. FP($B$)–{$\delta$} sera noté $F$ pour simplifier les écritures.

Pour le type d'assertion $\delta$, on montre par l'absurde que si l'assertion $\delta$ est déductible de F alors $\delta$ n'appartient pas à la forme première.

$c \leq c'$ : si $F \vdash c \leq c'$, $F \models c \leq c'$ et donc d'après le lemme 12, il existe une chaîne de $c$ à $c'$ : un ensemble de classes $c_1, \ldots, c_n$, telle que $c=c_1$, $c'=c_n$ et $\forall i \in [1..n\text{-}1]$ $(c_i \leq c_{i+1}) \in F$; comme $F \subseteq \mathrm{Cn}'(B)$, $\delta \notin \mathrm{FP}(B)$ par application de la règle de ($\leq$-réduction).

$o \in c$ : si $F \vdash o \in c$, $F \models o \in c$ et donc d'après le lemme 13, il existe $c'$ telle que $F \models c \leq c'$ et $(o \in c') \in F$, et comme $F \subseteq \mathrm{Cn}'(B)$, par application de la règle de ($\in$-réduction), $\delta \notin \mathrm{FP}(B)$.

$c.a \subseteq d$ si $F \vdash c.a \subseteq d$, $F \models c.a \subseteq d$ et donc d'après le lemme 14, soit $d$ est un domaine non minimal, soit $d = \bigwedge_{(c.a \subseteq d_i) \in F} d_i \wedge \bigwedge_{\substack{F \models c \leq c_j \\ (c_j.a \subseteq d_{jk}) \in F}} d_{jk}$.

Si $d$ est un domaine non minimal, il existe $F \vdash c.a \subseteq d'$ avec $d' <:_\tau d$ et dans ce cas, $\delta \notin$ FP($B$) d'après la règle de (d-normalisation).

Si $d = \bigwedge_{(c.a \subseteq d_i) \in F} d_i \wedge \bigwedge_{\substack{F \models c \leq c_j \\ (c_j.a \subseteq d_{jk}) \in F}} d_{jk}$, $d = \bigwedge_{\substack{(c \leq c_j) \in Cn'(B) \\ (c_j.a \subseteq d_{jk}) \in Cn'(B)}} d_{jk}$ car il n'existe pas d'assertion $c.a \subseteq d$ dans $F$ (vu que dans la forme première, il n'existe qu'une seule assertion de ce type pour $c$ et $a$ donnés et que cette assertion dans FP($B$) est $\delta$). Dans ce cas, par la règle de (d-anti-intersection), $\delta \notin$ FP($B$).

$o.a=v$ : si $F \vdash o.a=v$, $F \models o.a=v$ et donc d'après le lemme 15, il existe des classes $c_1, \ldots, c_n$, telles que $\forall i\ F \models o \in c_i$ et $F \models c_i.a \subseteq d_i$ et $\wedge_i \{d_i\} = \{v\}$ et comme $F \subseteq$ Cn'($B$), par application de la règle de (o-anti-valuation), $\delta \notin$ FP($B$). ◊

**Proposition 21 :**

$$\forall \delta \in \text{FP}(B) \text{ et } \forall B' \subseteq \text{Cn}(B) \text{ sous forme normale, } B' - \{\delta\} \nvdash \delta.$$

*démonstration.* la démonstration est exactement la même que la précédente, en posant $F = B' - \{\delta\}$. ◊

D'autre part, les propositions suivantes montrent que la forme première peut remplacer partout la clôture déductive (elle ne perd aucune information modulo la clôture déductive).

**Propositions 22 :**

(1) $B \vdash$ FP($B$)

(2) FP($B$) $\vdash B$

*démonstration.*

(1) $B \vdash$ Cn($B$) et FP($B$) $\subseteq$ Cn($B$).

(2) Chaque type d'assertion de $B$ est étudié pour montrer qu'une telle assertion peut être déduite à partir de FP($B$) :

$c \leq c'$ : si $(c \leq c') \in$ Cn($B$)–FP($B$) alors il a été supprimé par la règle de ($\leq$-réduction), c'est-à-dire qu'il existe une classe intermédiaire ($c \leq c''$ et $c'' \leq c' \in$ Cn($B$)); en procédant par récurrence, cette assertion est retirée dans la forme première s'il existe une chaîne de classes de $c$ à $c'$ et alors par application de la règle de ($\leq$-transitivité), FP($B$) $\vdash c \leq c'$.

$o \in c$ : si $(o \in c) \in$ Cn($B$)–FP($B$) alors il a été supprimé par la règle de ($\in$-réduction), c'est-à-dire qu'il existe $c'$ telle que ($c \leq c'$) et ($o \in c'$) appartiennent à Cn($B$). Or d'après le point précédent, FP($B$) $\vdash c \leq c'$ et en choisissant $c'$ la plus haute classe telle que $(o \in c') \in$ Cn'($B$), $(o \in c') \in$ FP($B$) (car cette assertion n'a pas pu être retirée de la forme première). Donc par application de la règle de ($\in$-clôture), FP($B$) $\vdash o \in c$.

$c.a \subseteq d$ : si $(c.a \subseteq d) \in$ Cn($B$)–FP($B$), le résultat est prouvé par récurrence sur la distance de la classe à la racine. Pour la classe racine, l'assertion a été retirée par la règle de (d-normalisation); il existe $(c.a \subseteq d') \in \mathrm{Cn}'(B)$ tel que $d' <:_\tau d$; soit l'assertion avec le domaine minimal (unique car dans $\mathrm{Cn}'(B)$, il y a toutes les intersection de domaines); cette assertion appartient à FP($B$) car elle ne peut pas être retirée de Cn($B$) car la (d-normalisation) ne s'appliquent pas et à partir d'elle, toutes les assertions de restriction de domaine peuvent être redéduites.

Soit à présent le cas d'une classe quelconque de la hiérarchie avec comme hypothèse de récurrence que pour toutes ses surclasses $\mathrm{FP}(B) \vdash c'.a \subseteq d$ si $c'.a \subseteq d \in \mathrm{Cn}(B)$. Cette assertion a pu être retirée par trois règles différentes :

(d-normalisation) : s'il existe $(c.a \subseteq d') \in \mathrm{Cn}'(B)$ tel que $d' <:_\tau d$, on choisit l'assertion avec le domaine minimal et celle-ci appartient à F ou a été supprimée par (d-anti-intersection); pour ce dernier cas, le résultat est obtenu en utilisant l'hypothèse de récurrence.

(d-anti-intersection) : il existe un ensemble $\{c_i.a \subseteq d_i \quad c \leq c_i\} \in \mathrm{Cn}'(B)$ avec $\bigwedge_i d_i = d$. En utilisant l'hypothèse de récurrence et la règle de (d-intersection) le nombre de fois nécessaires, $\mathrm{FP}(B) \vdash c.a \subseteq d$.

Le résultat est donc montré par récurrence.

$o.a=v$ : si $(o.a=v) \in \mathrm{Cn}(B) - \mathrm{FP}(B)$, alors $(c_i.a \subseteq d_i) \in \mathrm{Cn}'(B)$ et $(o \in c_i) \in \mathrm{Cn}'(B)$ pour un ensemble de $i$ tels que $\{v\} = \bigwedge_i d_i$. Des points précédents de la démonstration, $\mathrm{FP}(B) \vdash c_i.a \subseteq d_i$ et $\mathrm{FP}(B) \vdash o \in c_i$ et donc $\mathrm{FP}(B) \vdash o.a=v$ par la règle de (o-valuation).

Le résultat est donc montré pour toutes les assertions de la base. ◊

D'autres propriétés intéressantes de la forme première, justifiant son utilisation, sont données ci-dessous, notamment l'équivalence de l'égalité des clôtures déductives de deux bases et de l'égalité de leur forme première (proposition 26).

**Propositions 23 :**

Cn(FP($B$))=Cn($B$)

FP(Cn($B$))=FP($B$)

$\mathrm{FP}(B) \vdash \mathrm{Cn}(B)$

$\mathrm{Cn}(B) \vdash \mathrm{FP}(B)$

*démonstration.* Immédiat d'après la proposition 22. ◊

**Proposition 24 : stabilité**

FP(FP($B$)) = FP($B$)

*démonstration.* Cn(FP($B$))=Cn($B$) et l'application de FP à ces deux termes va en enlever les mêmes éléments. ◊

**Proposition 25 :**

$\forall B, B'$, les trois conditions suivantes sont équivalentes :

(1) $\mathrm{FP}(B') \vdash \mathrm{FP}(B)$

(2) $B' \vdash B$

(3) $\mathrm{Cn}(B) \subseteq \mathrm{Cn}(B')$

*démonstration.*

(1) $\Rightarrow$ (2) : $B' \vdash \mathrm{FP}(B')$, $\mathrm{FP}(B') \vdash \mathrm{FP}(B)$ et $\mathrm{FP}(B) \vdash B$, d'où $B' \vdash B$.

(2) $\Rightarrow$ (3) : $B \subseteq \mathrm{Cn}(B')$ car $B' \vdash B$ donc $\mathrm{Cn}(B) \subseteq \mathrm{Cn}(\mathrm{Cn}(B')) = \mathrm{Cn}(B')$.

(3) $\Rightarrow$ (1) : $\mathrm{Cn}(B) \subseteq \mathrm{Cn}(B')$, $\mathrm{FP}(B') \vdash \mathrm{Cn}(B')$ donc $\mathrm{FP}(B') \vdash \mathrm{Cn}(B)$ et $\mathrm{FP}(B') \vdash \mathrm{FP}(B)$.◊

**Proposition 26 :**

$\forall B, B'$, les trois conditions suivantes sont équivalentes :

(1) $\mathrm{FP}(B) = \mathrm{FP}(B')$

(2) $B \vdash B'$ et $B' \vdash B$

(3) $\mathrm{Cn}(B) = \mathrm{Cn}(B')$

*démonstration.*

(2) $\Leftrightarrow$ (3) : immédiat d'après la proposition précédente.

(3) $\Rightarrow$ (1) : immédiat vu la définition de la forme première.

(1) $\Rightarrow$ (3) : immédiat d'après la proposition précédente. ◊

**Proposition 27 :**

La forme première d'une base $B$ est le plus petit ensemble permettant de déduire $B$ :

$\forall B'$ telle que $\mathrm{FP}(B') \neq \mathrm{FP}(B)$ et que $B' \vdash B$, $\mathrm{Cn}(\mathrm{FP}(B)) \subset \mathrm{Cn}(B')$.

*démonstration.* soit $B'$ telle que $\mathrm{Cn}(\mathrm{FP}(B)) \subset \mathrm{Cn}(B')$ soit faux; si $\mathrm{Cn}(\mathrm{FP}(B)) = \mathrm{Cn}(B')$, alors d'après la proposition 26, $\mathrm{FP}(\mathrm{FP}(B)) = \mathrm{FP}(B')$, c'est-à-dire $\mathrm{FP}(B) = \mathrm{FP}(B')$ d'après la proposition 24; si $\mathrm{Cn}(\mathrm{FP}(B)) \neq \mathrm{Cn}(B')$, $\mathrm{Cn}(\mathrm{FP}(B)) = \mathrm{Cn}(B) \not\subset \mathrm{Cn}(B')$ donc $B' \nvdash B$. ◊

L'intérêt plus pratique de cette forme première (ou d'autres formes normalisées adaptées aux représentations par objets) est la possibilité de l'obtenir facilement (par son calcul à partir d'une base quelconque) et de la manipuler de façon efficace pour déterminer la consistance, la cohérence ou une conséquence particulière d'une base. Pour le calcul de la forme première, il faut remarquer que les retraits peuvent être faits séquentiellement, en particulier pour les assertions concernant les restrictions d'attribut de domaine; en effet, commencer par normaliser réduit ces assertions à une par classe; la (d-anti-intersection) peut alors s'appliquer sur ce nombre restreint d'assertions afin de supprimer les dernières assertions redondantes.

De plus, pour déduire une assertion particulière (ou vérifier qu'une assertion particulière est déductible), les procédures de déductions peuvent être guidées de manière à aboutir rapidement à une conclusion car il suffit de déclencher :

*les règles* qui sont utiles pour le type de l'assertion à déduire (ainsi, pour déduire une assertion du type $c \leq c'$, nul besoin des règles de (d-héritage) ou de ($\in$-clôture))

*sur des assertions* dont les composants (objets ou classes) sont utiles en fonction de leur place respective par rapport à l'ordre de spécialisation entre les classes (par exemple, pour déduire une assertion du type $o \in c$, seules les assertions portant sur les classes supérieures à $c$ (par rapport à $\leq$) peuvent être utiles dans les règles d'inférence).

La conception et l'évaluation de tels algorithmes n'ont pas été complètement réalisées pour l'instant mais il semble raisonnable de penser que pour un langage du type de celui présenté ici, il est possible de développer des algorithmes polynômiaux.

Au delà de cet aspect algorithmique, la forme première est utilisée dans la définition de la révision.

# 2. Révision

Pour le langage d'objet dont la syntaxe et la sémantique ont été définies, il s'agit à présent d'étudier les possibilités de réviser une base de connaissance quand un ajout est inacceptable dans cette base. La définition de bases révisées proposée ici correspond à celle donnée généralement, mais elles sont exprimées sous une forme particulière utilisant la forme première (§2.1). L'une des particularités de cette étude est de déterminer un ensemble de bases révisées et non une seule; le critère de choix entre les différentes bases révisées possibles n'étant pas formalisé, ce choix est laissé à l'utilisateur de la base de connaissance. Cependant, il s'agit de restreindre au maximum les possibilités de ce choix en ne proposant que les bases qui modifient le moins la base initiale. Cette modification minimale est formalisée par la minimalité sémantique (la proximité, §2.2) et la minimalité syntaxique définie à l'aide d'une relation d'ordre entre bases révisées, relation dépendant des assertions supprimées et ajoutées dans la base (§2.3). L'équivalence de ces deux minimalités est montrée et une caractérisation syntaxique d'une base révisée minimale est donnée (§2.4).

## 2.1. Base révisée

La révision va agir sur des bases acceptables, c'est-à-dire qui correspondent à des bases qui ne contiennent ni connaissances contradictoires, ni classes inutilisables (définition 11).

**Définition 20 : base révisée**

Soient $B$ une base acceptable et $\delta$ une assertion telle que $B\cup\{\delta\}$ soit inacceptable. $B'$ est une base révisée de $(B,\delta)$ ssi $B'\cup\{\delta\}$ est acceptable et $Cn(B')\subset Cn(B)$ (ce qui se traduit sémantiquement par $B\models B'$).

La notion de révision présentée ici (la base révisée étant $B'\cup\{\delta\}$) satisfait les six premiers postulats de [1], ainsi que ceux proposés dans [10, §7.1.2]. Elle correspond donc à la notion de révision couramment utilisée (à la différence qu'elle traite également les cas d'incohérence). En cas d'inconsistance ou d'incohérence, elle permet de se replier sur une base $B'$ plus réduite que $B$ mais compatible avec l'ajout de $\delta$.

Les bases révisées vont être exprimées sous une forme plus adaptée à leur manipulation. Intuitivement, un certain nombre d'assertions ($B'_-$) sont supprimées de la forme première — qui ne seront effectivement plus déductibles grâce à la minimalité de la forme première (proposition 20) — et certaines assertions ($B'_+$), déductibles de $B$ mais pas de $FP(B)-B'_-$, sont cependant conservées. Pour avoir une forme plus simple, l'intersection de ces deux ensembles est vide (sinon des éléments seraient retranchés et ré-ajoutés).

**Proposition 28 :**

Si $B'$ est une base révisée de $(B,\delta)$, alors $B'$ peut se caractériser par l'écriture suivante : $B' = (FP(B)-B'_-)\cup B'_+$ avec comme contraintes :

- $B'_- \cap B'_+ = \emptyset$,
- $B'_- \subseteq FP(B)$ et
- $B'_+ \subseteq Cn(B)$.

*démonstration*. Soit $B'$ une base révisée; déterminons $B'_-$ et $B'_+$.

$B'_- = FP(B) - FP(B')$

$B'_+ = B'-(FP(B)-B'_-)$

On a bien : $B' = (FP(B)-B'_-)\cup B'_+$, $B'_-\subseteq FP(B)$ et $B'_+\subseteq Cn(B)$ (car $B'\subseteq Cn(B)$)

De plus, $B'\cap B'_-=\emptyset$, donc $B'_+\cap B'_-=\emptyset$. ◊

La figure 6 explicite ce que contient une base révisée en fonction de $B'_+$ et $B'_-$.

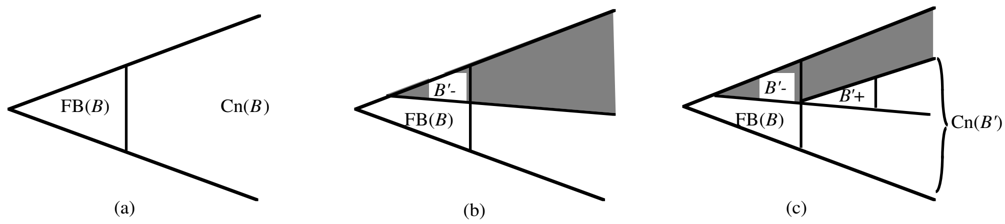


Figure 6 : la forme première $FP(B)$ permet de déduire la clôture déductive $Cn(B)$ (a); le retrait des assertions appartenant à $B'_-$ ne rend plus possible la déduction d'un certain nombre d'assertions de $Cn(B)$ (b); parmi ces assertions, l'ensemble $B'_+$ est ajouté et permet alors de redéduire une partie des assertions qui avaient été supprimées par le retrait de $B'_-$ (c).

Jusqu'ici les bases révisées ne sont pas contraintes à être exprimées sous forme première : $B'_+$ peut contenir des informations déductibles de FP($B$)$-B'_-$ ou des informations redondantes. Différentes bases révisées peuvent donc avoir la même clôture déductive et donc la même forme première (figure 7).

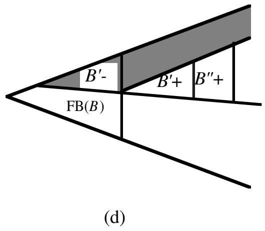


Figure 7 : les bases révisées définies par $B'$ = (FP($B$)$-B'_-$)$\cup B'_+$ et $B''$= (FP($B$)$-B'_-$)$\cup B''_+$ ont la même clôture déductive.

Dans la suite, on ne considérera plus que l'ensemble des bases révisées quotienté par la relation d'équivalence «avoir la même image par Cn (ou par FP)». L'expression «base révisée» désignera donc la représentante sous forme première d'une classe de bases révisées. Une base révisée est donc caractérisée *de manière unique* par : $B'$ est sous forme première et $B'$=(FP($B$)$-B'_-$)$\cup B'_+$. Dans ce cas, $B'_+ \subseteq$Cn($B$)–Cn(FP($B$)$-B'_-$)$-B'_-$ et dans $B'_+$ il n'y a que les assertions strictement nécessaires (celles qui étaient redondantes avec d'autres assertions de Cn(FP($B$)$-B'_-$) ne sont pas autorisées et il n'y a pas d'assertions redondantes dans $B'_+$, vu que la base révisée est exprimée sous forme première).

**Exemple :**

La figure 8 représente graphiquement différentes bases révisées possibles de la base inconsistante de la figure 2 :

FP($B$)={$c_1 \leq c_0$ ; $c_2 \leq c_1$ ; $c_3 \leq c_2$ ; $c_4 \leq c_2$ ; $i \in c_2$ ; $i' \in c_3$ ; $i'' \in c_4$ ; $c_0.a \subseteq$[-∞ +∞] ; $c_1.a \subseteq$[0 100] ; $c_2.a \subseteq$[10 25] ; $c_4.a \subseteq$[10 15] ; $i.a$=24 ; $i'.a$=18} avec $\delta$=($i''.a$=7).

Selon la notation donnée, ces différentes bases révisées (exprimées sous forme première) se détaillent comme suit :

$B_1$ = FP($B$) $-$ {$i'' \in c_4$} $\cup$ {$i'' \in c_1$}

$B_2$ = FP($B$) $-$ {$i'' \in c_4$} $\cup$ {$i'' \in c_0$}

$B_3$ = FP($B$) $-$ {$c_2.a \subseteq$[10 25]; $c_4.a \subseteq$[10 15]} $\cup$ {$c_2.a \subseteq$[7 25]; $c_4.a \subseteq$[7 15]}

$B_4$ = FP($B$) $-$ {$c_2.a \subseteq$[10 25]; $c_4.a \subseteq$[10 15]} $\cup$ {$c_2.a \subseteq$[5 25]; $c_4.a \subseteq$[5 15]}

$B_5$ = FP($B$) $-$ {$c_4 \leq c_2$; $c_4.a \subseteq$[10 15]} $\cup$ {$c_4 \leq c_0$; $c_4.a \subseteq$[7 15]}

$B_6$ = FP($B$) $-$ {$c_4 \leq c_2$; $c_4.a \subseteq$[10 15]} $\cup$ {$c_4 \leq c_1$; $c_4.a \subseteq$[7 15]}

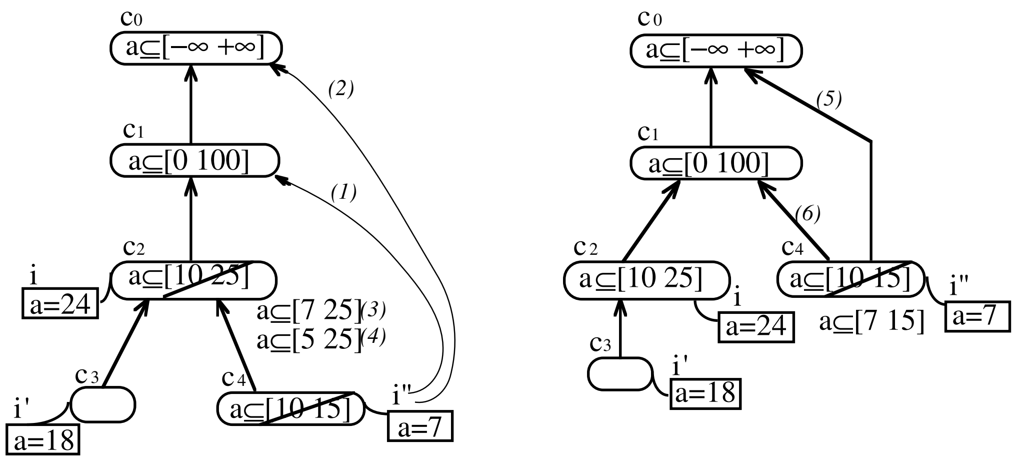


Figure 8 : exemples de bases révisées possibles : (1) et (2) : l'instance est déplacée sous une autre classe (suppression de son appartenance à sa classe d'origine et attachement à l'une de ses surclasses (puisque les ajouts sont à faire dans Cn($B$))). (3) et (4) : deux assertions de restriction de domaine est supprimée (elle peuvent être remplacées par une autre — le domaine étant plus grand que le précédent). (5) et (6) : la classe est déplacée en supprimant une assertion de spécialisation et en la remplaçant par d'autres et sa restriction de domaine est modifiée.

D'autres bases révisées existent (comme FP($B$)−$\{i''\in c_4\ ;c_4\leq c_2\}$), mais elles ne correspondent pas à des bases qui modifient minimalement la base initiale. Dans la littérature, différentes définitions de la minimalité de la révision logique sont utilisées : le minimum d'atomes changent de valeur [7]; la base révisée correspond à un sous-ensemble maximalement consistant de la base initiale et préserve les connaissances indiquées comme les plus importantes en utilisant une relation d'ordre [1,11]. La minimalité employée ici est précisée grâce aux définitions des notions de proximité (d'un point de vue sémantique) et de minimalité (d'un point de vue syntaxique). Nous montrerons que ces deux notions coïncident. À noter que cette relation entre bases révisées possibles ne permet pas généralement de définir une unique base révisée minimale.

## 2.2. Proximité

Lors de la révision, la base de connaissance doit être modifiée de manière à ce qu'elle devienne acceptable tout en s'éloignant le moins possible de la base initiale. Cette notion s'exprime sémantiquement par la notion de proximité.

**Définition 21 : base révisée la plus proche possible**

$B'$ est une base révisée la plus proche possible de $(B,\delta)$ ssi

$$\forall\gamma \text{ tel que } B\models\gamma \text{ et } B'\cup\{\delta\}\not\models\gamma \text{ alors } B'\cup\{\delta\}\cup\{\gamma\} \text{ est inacceptable.}$$

Intuitivement $B'$ est (une base révisée) la plus proche possible de $(B,\delta)$, au sens où aucune autre base révisée de $(B,\delta)$ ne permet de satisfaire plus (au sens de l'inclusion ensembliste) d'assertions de $B$.

## 2.3. Minimalité

La notion de minimalité (des changements opérés sur la base) peut être définie par un ordre sur les ensembles de modifications opérées.

**Définition 22 : ordre sur les bases révisées**

Soient $B'$ et $B''$ deux bases révisées de $(B,\delta)$, $B'\propto B''$ ssi $\mathrm{Cn}(B'')\subseteq\mathrm{Cn}(B')$.

Intuitivement, une base révisée $B'$ est plus petite qu'une base $B''$ si $B'$ permet de déduire plus d'assertions (appartenant à la base initiale, car $\mathrm{Cn}(B')\subseteq\mathrm{Cn}(B)$ d'après la définition de base révisée)

**Exemple :**

Dans l'exemple de base inconsistante précédent, on a donc :

$B_1\propto B_2$ (d'après le troisième point de la proposition),

$B_3\propto B_4$ (d'après le second point de la proposition) et

$B_6\propto B_5$, (d'après les premier et second points de la proposition).

Les autres bases révisées ne sont pas comparables.

Cet ordre peut aussi s'exprimer à l'aide de l'écriture caractéristique de base révisée définie plus haut (proposition 28).

**Proposition 29 :**

Si $B'$ et $B''$ sont deux bases révisées de $(B,\delta)$, $B'$ est plus petite que $B''$ ssi $B'_-\subseteq B''_-$ et $B'\vdash B''_+$.

*démonstration.*

si) $B'_-\subseteq B''_- \Rightarrow \mathrm{FP}(B)\text{-}B''_-\subseteq\mathrm{FP}(B)\text{-}B'_-$

$\Rightarrow \mathrm{FP}(B)\text{-}B''_-\subseteq\mathrm{FP}(B)\text{-}B'_-\cup B'_+$

$\Rightarrow \mathrm{FP}(B)\text{-}B''_-\subseteq\mathrm{Cn}(\mathrm{FP}(B)\text{-}B'_-\cup B'_+)=\mathrm{Cn}(B')$

De plus, $B'\vdash B''_+$ donc $B''_+\subseteq\mathrm{Cn}(B')$

Donc, $\mathrm{FP}(B)\text{-}B''_-\cup B''_+\subseteq\mathrm{Cn}(B')$ ce qui équivaut à $\mathrm{Cn}(B'')\subseteq\mathrm{Cn}(B')$.

seulement si)

- $\mathrm{Cn}(B'')\subseteq\mathrm{Cn}(B')\Rightarrow B''_+\subseteq\mathrm{Cn}(B')$ car $B''_+\subseteq B''\subseteq\mathrm{Cn}(B'')$.
- Il faut montrer par l'absurde que $B'_-\subseteq B''_-$ : soit $\delta$ tel que $\delta\in B'_-$ et $\delta\notin B''_-$. Vu que $\delta\notin B''_-$ et $\delta\in\mathrm{FP}(B)$ (par définition), $\delta\in\mathrm{Cn}(B'')$. De plus, $B'$ est sous forme première donc d'après la proposition 27, $B'\nvdash\delta$. Ainsi, $\delta\in\mathrm{Cn}(B'')$ et $\delta\notin\mathrm{Cn}(B')$, d'où la contradiction et donc $B'_-\subseteq B''_-$. ◊

Intuitivement, $B'$ est plus petite que $B''$ si $B'$ a moins d'assertions supprimées et qu'elle permet de déduire tout ce qui a été rajouté dans $B''$.

**Proposition 30 :**

Soient $B'$ et $B''$ deux bases révisées de $(B,\delta)$, $B'\propto B''$ ssi $B'\models B''$.

*démonstration*. Immédiat en se servant de l'équivalence entre $B'\models B''$ et $\mathrm{Cn}(B'')\subseteq\mathrm{Cn}(B')$. ◊

Cet ordre permet de guider la recherche des bases révisées. Bien entendu, ce travail reste complexe puisque la relation d'ordre fait appel à la clôture déductive (même dans le début d'analyse ci-dessous).

**Définition 23 : base révisée minimale**

$B'$ est une base révisée minimale de $(B,\delta)$ ssi $B'$ est une base révisée de $(B,\delta)$ et qu'il n'existe pas $B''\neq B'$ telle que $B''$ soit une base révisée de $(B,\delta)$ et $B''\propto B'$.

## 2.4. Précisions sur la minimalité définie

La proximité qui définit sémantiquement ce qu'est une base révisée minimale et la minimalité définie syntaxiquement sont équivalentes.

**Proposition 31 : équivalence de la proximité et de la minimalité**

$B'$ est une base révisée minimale de $(B,\delta)$ ssi $B'$ est une base révisée la plus proche possible de $(B,\delta)$.

*démonstration.*

seulement si) par l'absurde, $B'$ est minimale et il existe $\gamma$ tel que $B\models\gamma$ et $B'\cup\{\delta\}\not\models\gamma$ et $B'\cup\{\delta\}\cup\{\gamma\}$ est acceptable. Soit $B''=B'\cup\{\gamma\}$; $B''$ est une base révisée car $B''\cup\{\delta\}$ est acceptable et $\mathrm{Cn}(B'')\subseteq\mathrm{Cn}(B)$. De plus, $B''\models B'$ et $B'\not\models B''$, donc $B''\propto B'$, donc $B'$ n'est pas minimale, d'où la contradiction.

si) par l'absurde, $B'$ est une base révisée la plus proche possible de $(B,\delta)$ et il existe $B''\neq B'$ telle que $B''$ est une base révisée de $(B,\delta)$ et $B''\propto B'$. Comme $\mathrm{Cn}(B')\subset\mathrm{Cn}(B'')$, il existe $\gamma$ tel que $B''\models\gamma$ et $B'\cup\{\delta\}\not\models\gamma$ alors $B'\cup\{\delta\}\cup\{\gamma\}$ est inacceptable et comme $B'\cup\{\delta\}\cup\{\gamma\}\subset\mathrm{Cn}(B'')$, $B''$ est inacceptable d'où la contradiction. Si $\forall\gamma$ tel que $B''\models\gamma$ alors $B'\cup\{\delta\}\models\gamma$, alors $B'\cup\{\delta\}\models B''\cup\{\delta\}$ et $B''\cup\{\delta\}\models B'\cup\{\delta\}$ et comme $B'$ et $B''$ sont exprimées sous forme première $B'=B''$, il y a contradiction. ◊

Les définitions précédentes (définition 21 et 23) précisent ce qu'est la notion de minimalité de manière globale, mais il est intéressant de savoir quelles sont les modifications effectuées sur la base. Le théorème suivant donne une caractérisation syntaxique d'une base révisée minimale. Elle indique qu'une base est une base révisée minimale si :

aucune autre base révisée ne fait moins de suppressions et plus d'ajouts;

toute classe est attachée à la plus basse classe possible;

toute instance est attachée à la plus basse classe possible;

toute restriction de domaine est effectuée dans la plus haute classe possible et avec un domaine le plus réduit possible (mais qui contient la restriction de domaine initiale).

Ce théorème est important en particulier pour pouvoir déterminer de manière effective si les solutions proposées par un algorithme sont minimales.

**Théorème 32 :**

$B'$ est une base révisée minimale de $(B,\delta)$ ssi :

(0) $B'$ est une base révisée de $(B,\delta)$.

(1) $\forall B''$ base révisée de $(B,\delta)$, si $B''_{-} \subseteq B'_{-}$ et $B'_{+} \subseteq B''_{+}$, alors $B'=B''$.

(2) $\forall(c \le c') \in B'_{+}$, $\forall c_1 \neq c'$, si $B' \vdash c_1 \le c'$ et $(c \le c_1) \in \mathrm{Cn}(B)$, alors $(c \le c_1)$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$.

(3) $\forall(o \in c) \in B'_{+}$, $\forall c' \neq c$, si $B' \vdash c' \le c$ et $(o \in c') \in \mathrm{Cn}(B)$, alors $(o \in c')$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$.

(4) $\forall$ $(c.a \subseteq d) \in B'_{+}$, $\forall (c',d') \neq (c,d)$, si $d' \le : d$, $B' \vdash c \le c'$ et $(c'.a \subseteq d') \in \mathrm{Cn}(B)$, alors $(c'.a \subseteq d')$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$.

Les deux lemmes suivants seront utiles pour la démonstration du "si" du théorème.

**Lemme 33 :**

Soit $B'$ une base révisée de $(B,\delta)$ vérifiant les conditions (0), (1) et (2); s'il existe $c$, $c'$ tels que $B \vdash c \le c'$ et $B' \nvdash c \le c'$, alors $(c \le c')$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$.

*démonstration.*

Il existe deux cas distincts selon l'existence d'une assertion du type $(c \le c'')$ dans $B'_{+}$.

S'il existe une classe $c''$ telle que $(c \le c'') \in B'_{+}$, alors :

si $\mathrm{FP}(B) - B'_{-} \vdash c' \le c''$ $(c' \neq c'')$, alors $(c \le c'')$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$ d'après la condition (2) et donc $B'$ n'est pas une base révisée de $(B,\delta)$.

si $\mathrm{FP}(B) - B'_{-} \vdash c'' \le c'$ $(c' \neq c'')$, alors $B' \vdash c \le c'$ (par application de la règle de $\le$-transitivité), ce qui est contraire aux hypothèses.

si $\mathrm{FP}(B) - B'_{-} \nvdash c' \le c''$ et $\mathrm{FP}(B) - B'_{-} \nvdash c'' \le c'$ $(c' \neq c'')$, alors considérons la base définie par $B''_{-} = B'_{-}$ et $B''_{+} = B'_{+} \cup \{c \le c'\}$. Si $(c \le c')$ est une assertion acceptable dans $B' \cup \{\delta\}$, alors $B''$ est une base révisée car $B''$ est exprimée sous forme première (car il n'existe pas de lien entre les deux classes $c'$ et $c''$ et donc les assertions $(c \le c'')$ et $(c \le c')$ peuvent appartenir toutes deux à la forme première) et il y a contradiction avec la condition (1) donc $(c \leq c')$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$.

S'il n'existe pas de classe $c''$ telle que $(c \leq c'') \in B'_+$, soit la base définie par $B''_- = B'_-$ et $B''_+ = B'_+ \cup \{c \leq c'\}$; si $(c \leq c')$ est une assertion acceptable dans $B' \cup \{\delta\}$, alors $B''$ est une base révisée car $B''$ est exprimée sous forme première et dans ce cas, il y a contradiction avec la condition (1). ◊

**Lemme 34 :**

Soit $B'$ une base révisée de $(B,\delta)$ vérifiant les conditions (1) et (4); s'il existe *c, a* et *d* tels que $B \vdash c.a \subseteq d$ et $B' \nvdash c.a \subseteq d$, alors $(c.a \subseteq d)$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$.

*démonstration.*

La démonstration se décompose en deux points selon que dans les sous-classes il existe des restrictions de domaine déductibles de $(c.a \subseteq d)$ ou non.

S'il existe une classe $c'$ telle que $B' \vdash c' \leq c$ $(c' \neq c)$, $(c'.a \subseteq d') \in B'$ et $d \leq : d'$ alors $(c'.a \subseteq d') \in B'_+$ (car $(c'.a \subseteq d') \notin \mathrm{FP}(B)$, car $d \leq : d'$ et $(c'.a \subseteq d) \in \mathrm{FP}(B)$ car $B \vdash c.a \subseteq d$ et $B \vdash c' \leq c$) et en utilisant la condition (4), $(c.a \subseteq d)$ n'est pas une assertion acceptable dans $B' \cup \{\delta\}$.

Si pour toute classe $c'$ telle que $B' \vdash c' \leq c$ $(c' \neq c)$, $(c'.a \subseteq d') \notin B'$ ou $(c'.a \subseteq d') \in B'$ et $\neg d \leq : d'$.

si $(c.a \subseteq d') \in B'_+$:

si $d' \leq : d$ alors $B' \vdash c.a \subseteq d$, ce qui est contraire aux hypothèses;

si $d < : d'$ alors $(c.a \subseteq d)$ non acceptable d'après la condition (4);

si $d'$ et $d$ sont non comparables, alors la base définie par $B''_- = B'_-$ et $B''_+ = B'_+ \cup \{c.a \subseteq d\}$ est sous forme première et est donc une base révisée si $(c.a \subseteq d)$ est acceptable; or dans ce cas, d'après la condition (1), $B' = B''$, ce qui est contradictoire avec les hypothèses.

s'il n'existe pas d'assertion $(c.a \subseteq d') \in B'_+$, la base définie par $B''_- = B'_-$ et $B''_+ = B'_+ \cup \{c.a \subseteq d\}$ est sous forme première (car les assertions de restriction de domaine dans les sous-classes peuvent toujours appartenir à la forme première) et est donc une base révisée si $(c.a \subseteq d)$ est acceptable; or dans ce cas, d'après la condition (1), $B' = B''$, ce qui est contradictoire aux hypothèses (car $B' \nvdash c.a \subseteq d$). ◊

*démonstration du théorème.*

La première partie de la démonstration est relativement immédiate; la seconde nécessite plus de moyens.

seulement si) $B'$ est une base révisée minimale de $(B,\delta)$.

(0) est immédiat.

(1) Si $B''$ est une base révisée de $(B,\delta)$, $B''_{-}\subseteq B'_{-}$ et $B'_{+}\subseteq B''_{+}$, alors $B''\vdash B'_{+}$, donc $B''\propto B'$, et comme $B'$ est une base minimale, $B''=B'$.

(2) Par l'absurde, supposons que $(c\leq c_1)$ soit acceptable dans $B'\cup\{\delta\}$ avec $B'\vdash c_1\leq c'$, $(c\leq c')\in B'_{+}$ et $c_1\neq c'$. La base $B''$ telle que $B''_{-}=B'_{-}$ et $B''_{+}=B'_{+}\cup\{c\leq c_1\}-\{c\leq c'\}$ est telle que $B''\cup\{\delta\}$ est acceptable et $B''$ est donc une base révisée; de plus, $B''\vdash B'_{+}$ (car $B''\vdash c\leq c'$, vu que $(c\leq c_1)\in B''_{+}$ et $B''\vdash c_1\leq c'$) et $B''\neq B'$ car $c_1\neq c'$, donc $B'$ n'est pas minimale, ce qui est contradictoire avec l'hypothèse.

(3) Par l'absurde, supposons que $(o\in c')$ soit acceptable dans $B'\cup\{\delta\}$ avec $B'\vdash c'\leq c$, $(o\in c)\in B'_{+}$ et $c\neq c'$. La base $B''$ telle que $B''_{-}=B'_{-}$ et $B''_{+}=B'_{+}\cup\{o\in c'\}-\{o\in c\}$ est sous forme première; $B''\cup\{\delta\}$ est acceptable et $B''$ est donc une base révisée; de plus, $B''\vdash B'_{+}$ (car $B''\vdash o\in c$, vu que $(o\in c')\in B''_{+}$ et $B''\vdash c'\leq c$) et $B''\neq B'$ car $c\neq c'$, donc $B'$ n'est pas minimale, ce qui est contradictoire avec l'hypothèse.

(4) Par l'absurde, si $(c'.a\subseteq d')$ est acceptable dans $B'\cup\{\delta\}$ avec $(c.a\subseteq d)\in B'_{+}$, $B'\vdash c\leq c'$ et $d'<:_{\tau}d$. La base $B''$ telle que $B''_{-}=B'_{-}$ et $B''_{+}=B'_{+}\cup\{c'.a\subseteq d'\}-\{c.a\subseteq d\}$ est telle que $B''\cup\{\delta\}$ est acceptable et $B''$ est donc une base révisée; de plus, $B''\vdash B'_{+}$ (car $B''\vdash c.a\subseteq d$, vu que $(c'.a\subseteq d')\in B''_{+}$, $B'\vdash c\leq c'$ et $d'<:_{\tau}d$) et $B''\neq B'$ car $(c',d')\neq(c,d)$, donc $B'$ n'est pas minimale, ce qui est contradictoire avec l'hypothèse.

si) Soit $B''$ telle que $B''\propto B'$ avec $B'$ vérifiant les cinq conditions, on va montrer que $B''=B'$. $B''\propto B'$, donc $B''_{-}\subseteq B'_{-}$ par définition. Si $B'_{+}\not\subset B''_{+}$, alors il existe $\mu\in B'_{+}$ et $\mu\notin B''_{+}$. $\mu\notin B''$ (car $\mu\notin B''_{+}$ et $\mu\notin(\mathrm{FP}(B)-B''_{-})$ car $B'_{+}\cap\mathrm{FP}(B)=\emptyset$ et $\mu\in B'_{+}$) et $B''\vdash\mu$. Dans la suite de la démonstration, on montre que l'existence d'une telle assertion $\mu$ conduit à des contradictions, en découpant la démonstration selon le type d'assertion qu'est $\mu$. Une fois cette démonstration faite, $B'_{+}\subseteq B''_{+}$ sera prouvé et comme $B''_{-}\subseteq B'_{-}$, donc $B'=B''$ d'après la condition (1).

Soit $\mu$ tel que $\mu\in B'_{+}$, $\mu\notin B''$ et $B''\vdash\mu$.

$c\leq c'$ : $(c\leq c')\in B'_{+}$, donc pour toute chaîne $c_1,\ldots,c_n$ de classes, telle que $c=c_1$, $c'=c_n$ et $\forall i\in[1..n\text{-}1]$ $(c_i\leq c_{i+1})\in\mathrm{FP}(B)$ (car $(c\leq c')\in\mathrm{Cn}(B)$), il existe un ensemble $I$ d'indices $\{i\}$ tel que $(c_i\leq c_{i+1})\in B'_{-}$ (sinon $(c\leq c')$ est déductible de $\mathrm{FP}(B)-B'_{-}$ et ne peut donc pas appartenir à $B'_{+}$). De plus, $B''\vdash c\leq c'$ et $(c\leq c')\notin B''$, donc il existe pour chaque $i\in I$, $(c_j\leq c_k)\in B''$ avec $1\leq j\leq i$ et $i+1\leq k\leq n$ (sachant qu'il peut s'agir de la même assertion pour plusieurs $i$).

S'il existe une assertion $(c_j\leq c_k)$ telle que $B'\nvdash c_j\leq c_k$, alors $(c_j\leq c_k)$ est inacceptable d'après le lemme 33 car $B\vdash c_j\leq c_k$.

Si pour toute les assertions ($c_j \leq c_k$), $B' \vdash c_j \leq c_k$, alors $B' \vdash c_j \leq c'$ (une chaîne est reconstituée entre les deux classes), $(c \leq c') \in B'_+$ :

si $c' \neq c_j$, ($c \leq c_j$) est inacceptable dans $B' \cup \{\delta\}$ d'après la condition (2); or $B'' \vdash B'$ et $B'' \vdash c \leq c_j$ d'où $B'' \cup \{\delta\}$ est une base inacceptable et $B''$ ne peut donc être une base révisée de $(B,\delta)$.

si $c' = c_j$, $B' \vdash c \leq c_k$ et comme $(c \leq c') \in B'_+$ et $c' \neq c_k$ (si $c' = c_k$, $(c \leq c') \in B''$, ce qui est contraire aux hypothèses), ($c \leq c_k$) est inacceptable dans $B' \cup \{\delta\}$ d'après la condition (2) et donc $B'' \vdash B'$ et $B'' \vdash c \leq c_j$ donc $B'' \cup \{\delta\}$ est une base inacceptable et ne peut être une base révisée de $(B,\delta)$.

$o \in c$ : $(o \in c) \notin B''$ et $B'' \vdash o \in c$, donc il existe au moins une classe $c' \neq c$ telle que $B'' \vdash c' \leq c$ et $(o \in c') \in B''_+$.

si $B' \vdash c' \leq c$, $(o \in c')$ n'est pas acceptable dans $B' \cup \{\delta\}$ (d'après la condition (3)); or $B'' \vdash B'$ et $B'' \vdash o \in c'$ donc $B'' \cup \{\delta\}$ est une base inacceptable et ne peut donc être une base révisée de $(B,\delta)$;

si $B' \nvdash c' \leq c$, ($c' \leq c$) n'est pas acceptable dans $B' \cup \{\delta\}$ (d'après le lemme 33); or $B'' \vdash B'$ et $B'' \vdash c' \leq c$ donc $B'' \cup \{\delta\}$ est une base inacceptable et ne peut donc être une base révisée de $(B,\delta)$.

$c.a \subseteq d$ : $(c.a \subseteq d) \in B'_+$ donc il existe un ensemble d'ensembles d'assertions $\{(c_i.a \subseteq d_i) \in \mathrm{FP}(B), B \vdash c \leq c_i\}$ avec $d = \wedge d_i$ (car $(c.a \subseteq d) \notin \mathrm{FP}(B)$); $B'' \vdash (c.a \subseteq d)$, mais $(c.a \subseteq d) \notin B''$, donc il existe au moins un ensemble $\{B'' \vdash c_i.a \subseteq d_i$ et $B'' \vdash c \leq c_i\}$ et pour cet ensemble, il existe $i$ tel que $B' \nvdash c_i.a \subseteq d_i$ ou $B' \nvdash c \leq c_i$ (sinon $(c.a \subseteq d)$ ne peut appartenir à la forme première de la base)

s'il existe $i$ tel que $B' \nvdash c \leq c_i$ alors ($c \leq c_i$) est inacceptable dans $B' \cup \{\delta\}$ (d'après le lemme 33) et donc, comme $B'' \vdash c \leq c_i$ et que $B'' \vdash B'$, $B'' \cup \{\delta\}$ est une base inacceptable et ne peut donc être une base révisée de $(B,\delta)$;

s'il existe $i$ tel que $B' \nvdash c_i.a \subseteq d_i$ alors $(c_i.a \subseteq d_i)$ est inacceptable dans $B' \cup \{\delta\}$ (d'après le lemme 34), $B'' \cup \{\delta\}$ est une base inacceptable et ne peut donc être une base révisée de $(B,\delta)$.

$o.a = v$ : $(o.a = v) \in B'_+$, donc il existe un ensemble d'ensembles d'assertions $\{B \vdash c_i.a \subseteq d_{ij}, (o \in c_i) \in B\}$ avec $\{v\} = \wedge d_{ij}$. Il existe au moins l'un de ces ensembles tel que $\{B'' \vdash (c_i.a \subseteq d_{ij}), B'' \vdash o \in c_i\}$ (car $(o.a = v) \notin B''_+$ et $B'' \vdash o.a = v$). Comme $(o.a = v) \in B'_+$, pour chacun de ces ensembles, il existe $i$ et $j$ tels que $B' \nvdash c_i.a \subseteq d_{ij}$ ou il existe $i$ tel que $B' \nvdash o \in c_i$ :

si $B' \nvdash c_i.a \subseteq d_{ij}$ alors $(c_i.a \subseteq d_{ij})$ est inacceptable dans $B' \cup \{\delta\}$ (d'après le lemme 34) et donc $B'' \cup \{\delta\}$ est une base inacceptable et ne peut donc être une base révisée de $(B,\delta)$.

si $B'\vdash c_i.a\subseteq d_{ij}$ et $B'\nvdash o\in c_i$ alors $(o\in c_i)$ est acceptable dans $B'\cup\{\delta\}$ car sinon $B''\cup\{\delta\}$ est une base inacceptable.

s'il existe une assertion $(o\in c)\in B'_+$ :

si $\mathrm{FP}(B)-B'_-\vdash c\leq c_i$ alors $B'\vdash o\in c_i$, ce qui est contraire aux hypothèses;

si $\mathrm{FP}(B)-B'_-\vdash c_i\leq c$ et $(c_i\neq c)$ alors $(o\in c_i)$ est inacceptable dans $B'\cup\{\delta\}$ (d'après la condition (3));

si $\mathrm{FP}(B)-B'_-\nvdash c\leq c_i$ et $\mathrm{FP}(B)-B'_-\nvdash c_i\leq c$, alors $B'''=B'\cup\{o\in c_i\}$ est sous forme première et est acceptable donc $B'''$ est une base révisée et d'après la condition (1), $B'''=B'$, ce qui n'est pas vrai car $B'\nvdash o\in c_i$.

s'il n'existe pas $c$ tel que $(o\in c)\in B'_+$, alors $B'''=B'\cup\{o\in c_i\}$ est sous forme première et est acceptable donc $B'''$ est une base révisée et d'après la condition (1), $B'''=B'$, ce qui n'est pas vrai car $B'\nvdash o\in c_i$.

Tous les cas possibles conduisent à une contradiction donc il n'existe pas d'assertion du type $(o.a=v)$ telle que $(o.a=v)\in B'_+$ et $(o.a=v)\notin B''_+$. ◊

La notion de minimalité définie dans ce travail peut être caractérisée de manière informelle par la préservation du maximum d'assertions qui étaient déductibles par la base initiale. Ceci se fait au détriment d'une autre idée qui est de supprimer les assertions préalablement déductibles d'assertions qui sont supprimées dans la base révisée. Un exemple simple permettant de voir la différence entre les deux approches est le suivant : si une valeur d'attribut est attribuée à une instance car cet attribut est restreint à un élément dans l'une de ses classes d'appartenance, et que la description d'attribut de la classe est modifiée dans la base révisée, cette valeur d'attribut d'instance sera tout de même préservée.

$B=\{c_1\leq c_0\,;\, i\in c_1\,;\, c_0.a\subseteq[10\ 30]\,;\, c_1.a\subseteq[30\ 100]\}$

$\mathrm{FP}(B)=\{c_1\leq c_0\,;\, i\in c_1\,;\, c_0.a\subseteq[10\ 30]\,;\, c_1.a\subseteq[30\ 30]\}$

$\mathrm{Cn}(B)=\{c_1\leq c_0\,;\, i\in c_1\,;\, c_0.a\subseteq[10\ 30]\,;\, c_1.a\subseteq[30\ 100]\,;\, c_1.a\subseteq[30\ 30]\,;\, i.a=30\}$

Si la révision consiste à supprimer l'assertion $c_1.a\subseteq[30\ 30]$, une base révisée minimale correspondante sera $B'=\{c_1\leq c_0\,;\, i\in c_1\,;\, c_0.a\subseteq[10\ 30]\,;\, i.a=30\}$, alors que $c_1.a\subseteq[30\ 30]$ n'est plus déductible.

Ce choix est fondé sur l'hypothèse suivante : l'utilisateur, en donnant explicitement certaines assertions, approuve également toutes celles qui pourront en être déduites et donc lors de la révision, il souhaite perdre le moins d'assertions possible parmi toutes celles qui étaient auparavant déductibles de la base.

Ainsi, la notion de base révisée minimale s'exprime syntaxiquement et sémantiquement. Dès lors, les procédures de révision qui peuvent être mises en œuvre syntaxiquement auront le sens de «conserver le maximum d'assertions valides». Ceci est possible pour l'instant grâce à la restriction du langage proposé. L'objectif des travaux futurs va donc être (a) de chercher à conserver ce résultat pour des langages étendus et (b) d'évaluer la complexité théorique et l'applicabilité pratique des techniques développées.

Les mécanismes de révision ont été peu développés, jusqu'à présent, pour des raisons de complexité. En effet, tous les moyens de réviser une base de connaissance ont besoin d'un oracle leur disant si la base est consistante ou non (dans la définition de minimalité donnée ci-dessus, la déduction est utilisée encore plus généralement). Si la complexité de cette décision est exponentielle, la révision est véritablement impraticable [11].

Dans le cas présenté ci-dessus, cette déduction devrait être polynomiale (et semble-t-il quadratique) dans le pire des cas et l'expérience pratique montre que les systèmes à objets (du moins ceux relativement simples : sans problème de satisfaction de contraintes, ni concepts récursifs ou restriction de rôles ouverts) permettent de décider rapidement de la consistance du système. Alors, l'idée de trouver de bonnes heuristiques pour la révision de bases d'objets vaut la peine d'être explorée; des algorithmes sont en cours de développement.

# 3. Retour aux objets

Les travaux présentés jusqu'à présent sont quelque peu éloignés des objets tels qu'ils sont habituellement considérés. Pour pouvoir envisager de les exploiter dans le cadre d'un système de représentation de connaissance concret, il faut disposer d'une articulation liant le système et le langage présenté ici (§3.1), ajouter un certain nombre d'expressions qui ne figurent pas dans le langage restreint et examiner si le résultat reste malgré tout praticable (§3.2).

## 3.1. Relation avec TROPES

Le langage qui a été présenté n'est pas celui utilisé par le système de représentation de connaissance TROPES [13] sur lequel la révision sera implémentée. Pour cela, la révision présentée ici devra être étendue à un langage plus riche et adaptée aux constructions de TROPES. La liaison entre les deux systèmes doit être détaillée et il faut donc considérer que le système TROPES a pour mission de s'assurer qu'il construit bien une base au sens énoncé plus haut.

En TROPES, les objets et les valeurs sont partitionnés en un ensemble de concepts et un ensemble de types. Aux valeurs correspond un domaine (qui sera $T_V$ dans le langage présenté ici) et aux objets un domaine (qui sera $T_O$). Les objets des concepts sont munis d'attributs, chaque concept constituant un espace de nom différent pour les noms de ces attributs. Ceux-ci sont transformés en un ensemble d'attributs $T_A$ dans le langage ci-dessus. Les concepts peuvent être munis de différents points de vue qui sont importants pour nommer les classes mais n'ont aucune contrepartie dans le langage ci-dessus. Chaque point de vue contient un ensemble de classes ordonnées en un arbre par la relation de spécialisation. L'espace de nommage des classes étant le point de vue, ici encore chaque classe aura un nom unique et l'ensemble des classes constituera donc $T_C$. L'ordre correspondant à la spécialisation sera rendu par des assertions de type ($c \leq c'$) et les contraintes posées sur les attributs dans les classes seront traduites par des assertions de type ($c.a \subseteq d$). La racine de chaque hiérarchie recevra donc aussi les contraintes portant sur les attributs du concept. Chaque objet dans $T_O$ fait partie d'un et d'un seul concept et est rattaché à une unique classe par point de vue ce qui est rendu par des assertions de type ($o \in c$). Il faut noter que l'attachement multiple n'a pas été prohibé jusqu'à maintenant. Enfin, les valeurs d'attributs des objets seront traduites par ($o.a{=}v$).

Voici ce que serait la base de l'exemple 1, exprimée dans le formalisme de TROPES.

```
<exemple-figure-1
  bases-requises = {} ;
  types = {"string", "integer"};
  points-de-vue = {};
  concepts = {
    <k
      clefs = {
        <name dans = string; nature = propriete; constructeur = un;>};
      attributs = {
       <a dans = integer; nature = propriete; constructeur = un;>};
      points-de-vue = {
        <v
          classe-racine =
            <c0 attributs = {<a>};>;
          classes = {
            <c1 super-classe = c0;
              attributs = {<a intervalles = {[0 100]};>};>,
            <c2 super-classe = c1;
              attributs = {<a intervalles = {[10 25]};>};>,
            <c3 super-classe = c2;
              attributs = {<a>};>,
            <c4 super-classe = c2;
              attributs = {<a intervalles = {[5 15]};>};>};>};
      passerelles = {};>};
  fichiers-instances = {
    <k name = "i"; est-dans = {c2@v}; a = 24;>,
    <k name = "i'"; est-dans = {c3@v}; a = 18;>,
    <k name = "i""; est-dans = {c4@v};>} ;
  modules-talk = {} ;
>
```

TROPES est capable de vérifier un certain nombre de propriétés dans la construction des bases : que les noms sont uniques, que les objets n'appartiennent qu'à un seul concept, que les valeurs et domaines des attributs sont bien typés, que les taxonomies sont des arbres, etc. Il est capable d'émettre des messages d'erreurs et de proposer des corrections immédiates suite à ces vérifications.

Une fois ce premier niveau de vérification passé, la base peut être considérée comme une base au sens énoncé ci-dessus et les mécanismes de révision pourront s'appliquer en retranchant ou ajoutant des assertions à la base TROPES. Le langage d'expression accepte aussi des bases qui ne sont pas des bases TROPES (par exemple, la spécialisation multiple n'a pas été prohibée) mais il suffit, pour détecter des inconsistances et proposer des révisions dans les bases TROPES, de pouvoir exprimer celles-ci dans le langage proposé et de ne pas tenir compte des bases révisées qui ne sont pas des bases TROPES.

## 3.2. Extensions

Le langage présenté ici est très restreint. Il n'est qu'une base à partir de laquelle des extensions pourront être ajoutées en surveillant les propriétés de la déduction et de la révision. Dans le cadre du système TROPES, les extensions suivantes devront être considérées :

Il faut pouvoir considérer divers types de données pas forcément aussi simples à manipuler que les entiers (TROPES autorise des types tels que les ensembles de dates). Par ailleurs, le langage d'expression des types est plus riche que les intervalles d'entiers (par exemple, TROPES autorise l'expression des unions d'intervalles pour les entiers). Cette extension pose un problème intéressant pour la révision (quelle est la révision minimale dans les unions d'intervalles ?) et risque d'amener un surcroît de complexité au problème.

La plus importante des extensions est celle qui autorise la valeur d'un attribut à être un objet et non une simple valeur. Le problème est délicat (en particulier, la complexité risque d'en pâtir) s'il n'y a pas de référence circulaire; il le devient encore plus lorsque les références circulaires sont autorisées (c'est-à-dire lorsqu'un objet fait référence à un autre objet qui le réfère à son tour).

Une extension plus typique de TROPES concerne l'ajout de passerelles (expressions s'interprétant comme $I[c_1]\cap\ldots\cap I[c_n]\subseteq I[c]$). Ce genre d'extension ne semble pas poser de problème particulier. Sa révision a été traitée en temps polynomial dans un cadre cependant plus restreint (pas de types ni d'attributs) [15].

Enfin, l'intégration à TROPES de problèmes de satisfaction de contraintes posera d'importants problèmes de complexité habituels dans ce contexte.

D'autre part, il sera intéressant d'automatiser le choix de la base révisée parmi l'ensemble des bases révisées minimales. Pour ce faire, des critères de préférence entre les connaissances devront être formalisés; cet ordre sur les connaissances ne doit cependant pas être trop fastidieux à définir, c'est pourquoi il est envisagé de se servir des différentes formes d'assertions. Ainsi, la construction d'une base de connaissance peut être décomposée en différentes étapes, pendant lesquelles soit les assertions reflétant la structure de la base ($c \leq c'$ ou $c.a \subseteq d$), soit les assertions concernant les instances ($o \in c$ ou $o.a = v$) sont privilégiées. Des fonctions de préférence pourront donc être définies pour sélectionner un sous-ensemble de bases révisées minimales, sans être forcément suffisamment précises pour obtenir une base révisée unique.

# 4. Comparaison avec d'autres travaux

Bernhard Nebel [10] a ouvert la voie des travaux concernant la révision dans les formalismes de représentation de connaissance dans un cadre différent de la logique. Mais la complexité du problème de la décision dans les systèmes considérés n'a pas permis de poursuivre. Ces travaux sont cependant précieux pour évaluer la complexité des problèmes à traiter [11].

Récemment Norman Foo [8] a proposé quelques méthodes pour réviser les bases de graphes conceptuels et de nombreux travaux ont étudié la réorganisation de bases d'objets (à l'aide d'invariants à respecter et de règles conservant ces invariants) [4,12,2]. Si ceux-ci permettent de se faire une idée des méthodes pour retrouver une base acceptable, ils ne considèrent pas explicitement la sémantique du formalisme de représentation de connaissance.

Dans de récents travaux, Theodorados [15] s'est attaqué au même problème que le travail présenté ici. Il part cependant de bases quelque peu différentes : le langage considéré est plus puissant dans l'expression des rapports entre les classes mais les attributs ne sont pas pris en compte. À partir d'une modélisation logique du langage de définition de classes, il définit deux procédures de révisions classiques. Il montre (pour des notions particulières de révision) la co-NP-complétude du problème général et des résultats de polynômialité dans plusieurs restrictions intéressantes pour TROPES.

# 5. Conclusion

Le travail présenté ici aborde la révision (ou la correction d'erreurs) dans une base de connaissance par objets sous un aspect formel. À cette fin, un langage de représentation de connaissance par objets restreint a été défini. Il a été doté d'une sémantique et d'un système déductif correct et complet. Sur cette base, la notion de révision a été définie. Cette révision est pourvue de critères de minimalité (syntaxique) et de proximité (sémantique) permettant d'ordonner les bases révisées. Ces deux notions coïncident, ce qui permet de définir des procédures valides sémantiquement sans faire appel aux ensembles de modèles.

Au delà du simple résultat formel, la notion de forme première d'une base de connaissance permet d'envisager des techniques de réalisation efficace de ces outils. Ce genre d'outil ne se cantonne pas à la représentation de connaissance. Il devrait être applicable à un langage orienté objet typé pour aider un concepteur, voire une application, à corriger les problèmes qui peuvent apparaître.

Trois perspectives permettent de baliser le développement futur de ce travail :

- une extension du langage utilisé et un aménagement des notions introduites;
- la mise en évidence de l'intérêt et de la faisabilité de cette approche grâce à l'implémentation;
- la définition de critères de préférence entre les connaissances.

Le but à terme est d'intégrer au système TROPES un mécanisme permettant, lors de la détection d'une erreur, d'indiquer à l'utilisateur les façons minimales permettant de corriger celle-ci. Nous pensons qu'un tel résultat est possible et qu'il peut être exporté vers d'autres formalismes de représentation de connaissance et de programmation par objets.

# Remerciements

Nous tenons à remercier Roland Ducournau pour avoir lu les premières (et les dernières) versions de ce document et contribué à l'améliorer notablement.

# Références

[1] C. Alchourrón, E. Gärdenfors, P. Makinson. *On the logic of theory change : partial meet contraction and revision functions*. Journal of symbolic logic. 50(2):510-530. 1985.

[2] J. Barnejee, W. Kim, H.J. Kim, H. Korth. *Semantics and implementation of schema evolution in object-oriented databases*. SIGMOD records. 16(3):311-322. 1987

[3] C. Barral, S. Krauss, J. Minker, V. Subrahmanian. *Combining knowledge bases consisting of first-order theories.* Computational Intelligence. 8(1):45-71. 1992.

[4] A. Borgida. *Language features for flexible handling of exceptions in informations systems.* ACM transactions on database system. 10(4):565-603. 1985.

[5] C. Capponi. *Identification et exploitation des types dans un modèle de connaissances à objets.* Thèse d'informatique. Université Joseph Fourier. Grenoble (FR). 1995.

[6] I. Crampé, J. Euzenat. *Révision interactive dans une base de connaissance à objets.* Actes 10ième RFIA. Rennes (FR). pp.615-623. 1996.

[7] M. Dalal. *Investigations into a theory of knowledge base revision : preliminary report.* Actes 7th AAAI. Philadelphia (PA US). pp.475-479. 1988.

[8] N. Foo. *Ontology revision*. Actes 'ICCS-95'. Santa Cruz (CA US). LNCS 954. 1995.

[9] M. Kifer, G. Lausen, J. Wu. *Logical foundations of object-oriented and frame-based languages*. Journal of the ACM 42(4):741-843. 1995.

[10] B. Nebel. *Reasoning and revision in hybrid representation systems*. LNCS 422. 1990.

[11] B. Nebel. *Syntax-based approaches to belief revision*. in P. Gärdenfors (éd.). Belief revision. Cambridge tracts in theoretical computer science 29. Cambridge university press. Cambridge (GB). pp.52-88. 1992.

[12] J. Penney, J. Stein. *Class modification in the GemStone object-oriented DBMS*. Actes 2nd OOPSLA. Orlando (FL US). pp.111-117. 1987.

[13] Projet Sherpa. *TROPES 1.0 reference manual*. Rapport interne INRIA Rhône-Alpes. Grenoble (FR). 1995.

`ftp://ftp.inrialpes.fr/pub/sherpa/rapports/tropes-manual.ps.gz`

[14] Léa Sombé (éds). *Special issue on Revision and Updating in Knowledge Base.* International journal of intelligent systems. 9(1). 1994.

[15] D. Theodoratos. *Updating object-oriented schema structures viewed as logical theories*. Rapport de recherche 12. ERCIM. Rocquencourt (FR). 1995.

# Notations

**Opérations sur les types**

| | |
|---|---|
| $d \wedge_\tau d'$ | intersection des domaines de valeurs $d$ et $d'$. |
| $d =_\tau d'$ ($d \neq_\tau d'$) | le domaine $d$ est égal au (resp. différent du) domaine $d$. |
| $d \leq:_\tau d'$ ($d <:_\tau d'$) | le domaine $d$ est inclus (resp. strictement) dans le domaine $d'$. |
| $v :_\tau d$ ($\neg v :_\tau d$) | la valeur $v$ appartient (resp. n'appartient pas) au domaine $d$. |
| $v \leq_\tau v'$ | la valeur $v$ est inférieure à la valeur $v'$ ($\tau$ étant ordonné). |
| $v =_\tau v'$ ($v \neq_\tau v'$) | la valeur $v$ est égale à (resp. différente de) la valeur $v'$. |
| $\perp_\tau$ | symbole représentant le domaine vide du type $\tau$. |

**Ensembles d'entités**

| | |
|---|---|
| $T_C^B$ | l'ensemble des classes $c$ introduites dans la base $B$ par une assertion de la forme $c \leq c'$ ou $c.a \sqsubseteq d$. |
| $T_O^B$ | l'ensemble des objets $o$ introduits dans la base $B$ par une assertion de la forme $o.a = v$ ou $o \in c$. |
| $T_A^B$ | l'ensemble des attributs $a$ introduits dans la base $B$ par une assertion de la forme $o.a = v$ ou $c.a \sqsubseteq d$. |
| $T_V^B$ | l'ensemble des valeurs des types introduits dans la base $B$ (ici les entiers). |
| $T_T^B$ | l'ensemble des domaines des types introduits dans la base $B$ (ici les intervalles d'entiers). |

**Assertions**

| | |
|---|---|
| $c \leq c'$ | la classe $c$ est sous-classe de la classe $c'$. |
| $o \in c$ | l'objet $o$ appartient à la classe $c$. |
| $c.a \sqsubseteq d$ | l'attribut $a$ de la classe $c$ est restreint au domaine $d$. |
| $o.a = v$ | la valeur de l'attribut $a$ de l'objet $o$ est $v$. |

**Interprétation et déduction**

| | |
|---|---|
| $\langle D, I \rangle$ | une interprétation est un couple composé d'un domaine et d'une fonction d'interprétation. |
| $\models_M B$ | $M$ est un modèle de la base $B$. |
| $B \models \delta$ | $\delta$ est une conséquence (sémantique) de $B$. |
| $B \vdash \delta$ | la base $B$ permet de déduire (syntaxiquement) l'assertion $\delta$. |
| $\mathrm{Cn}(B)$ | clôture déductive de la base par les règles d'inférence. |
| $\mathrm{FP}(B)$ | forme première de la base $B$. |