



**HAL**  
open science

## Elect : Une méthode de gestion des incohérences dans des ontologies légères partiellement préordonnées \*

Sihem Belabbes, Salem Benferhat, Jan Chomicki

### ► To cite this version:

Sihem Belabbes, Salem Benferhat, Jan Chomicki. Elect : Une méthode de gestion des incohérences dans des ontologies légères partiellement préordonnées \*. 13èmes Journées d'Intelligence Artificielle Fondamentale (JIAF 2019), Jul 2019, Toulouse, France. hal-02301980

**HAL Id: hal-02301980**

**<https://hal.science/hal-02301980v1>**

Submitted on 1 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Elect : Une méthode de gestion des incohérences dans des ontologies légères partiellement préordonnées \*

Sihem Belabbes<sup>1</sup> Salem Benferhat<sup>1</sup> Jan Chomicki<sup>2</sup>

<sup>1</sup> CRIL, Université d'Artois, CNRS-UMR8188, Lens, France

<sup>2</sup> SUNY at Buffalo, Buffalo, NY, USA

{belabbes,benferhat}@cril.fr chomicki@buffalo.edu

## Résumé

Nous nous intéressons au problème de la gestion des incohérences dans des ontologies légères. Nous supposons que les bases de connaissances terminologiques (TBox) sont spécifiées dans les logiques de description DL-Lite. Nous supposons aussi que les bases assertionnelles (ABox) sont partiellement préordonnées et peuvent être incohérentes avec la base terminologique TBox. Une contribution importante de cet article est la proposition d'une méthode efficace et saine, appelée Elect, pour restaurer la cohérence de la base ABox par rapport à la base TBox. Notre méthode permet de retrouver la sémantique dite IAR (*Intersection ABox Repair*) lorsque la base ABox est non-ordonnée. Elle permet également de retrouver la sémantique dite non-contestée lorsque la base ABox est totalement préordonnée. La justification sémantique de la méthode Elect est basée d'abord sur le fait qu'une base ABox partiellement préordonnée est interprétée comme une famille de bases ABox totalement préordonnées. Ensuite, la méthode Elect applique l'inférence non-contestée à chacune de ces ABox totalement préordonnées. Dans la deuxième partie de l'article, nous introduisons le concept d'assertions, dites élues, qui permet de donner une caractérisation équivalente de la méthode Elect sans avoir à générer explicitement toutes les bases ABox totalement préordonnées. Nous montrons alors que le calcul des assertions élues se fait en temps polynomial.

## Abstract

We focus on the problem of handling inconsistency in lightweight ontologies. We assume that terminological knowledge bases (TBoxes) are specified in DL-Lite and that assertional facts (ABoxes) are partially preordered and may be inconsistent with respect to TBoxes. One of the main contributions of this paper is the provision of an efficient and safe method, called Elect, to restore consistency of the ABox with respect to the TBox. In the case where the ABox

is flat (no priorities are associated with the assertions) or totally preordered, our method collapses with the well-known IAR semantics and non-defeated semantics, respectively. The semantic justification of Elect is obtained by first viewing a partially preordered ABox as a family of totally preordered ABoxes, and then applying non-defeated inference to each of the totally preordered ABoxes. We introduce the concept of elected assertions which allows us to provide an equivalent characterisation of Elect without explicitly generating all totally preordered ABoxes. Finally we show that the computation of Elect is done in polynomial time.

## 1 Introduction

Dans cet article, nous nous intéressons à la gestion des incohérences dans des ontologies spécifiées en DL-Lite [12], une famille de fragments légers des Logiques de Description (DL) ayant de bonnes propriétés calculatoires. Dans le contexte des Logiques de Description, une base de connaissances (KB) comporte deux composantes : une base TBox qui contient la connaissance terminologique et une base ABox qui est une base assertionnelle. Il est raisonnable et d'usage de supposer que le contenu de la base TBox est correct et sans conflits, donc les éléments de la base TBox ne sont pas remis en cause en présence de conflits. Cependant, les assertions de la base ABox peuvent être discutables lorsque la base de connaissances est incohérente. Plusieurs stratégies ont été proposées pour permettre de raisonner avec des bases de connaissances incohérentes [8, 13, 23] (voir également [2] pour un état de l'art). Cela revient souvent à calculer des réparations pour la base ABox, où une réparation est définie comme un sous-ensemble maximal de la base ABox qui est cohérent avec la base TBox.

L'approche dite *ABox Repair* (AR) [17] revient à réparer la base ABox de manière minimale (en termes d'inclusion

\*Cet article est paru en anglais dans les actes de 15<sup>th</sup> International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR), Philadelphie, USA, 3–7 Juin 2019, pages 210–223, Springer.

ensembliste) sans modifier la base TBox. La réponse à une requête est basée sur les réponses obtenues de chacune des réparations. L’approche AR est souvent considérée comme un moyen fiable de gérer les conflits. Cependant, le calcul des réponses aux requêtes avec la méthode AR est coûteux, même pour une logique d’ontologie légère telle que DL-Lite. La méthode connue sous le nom *Intersection ABox Repair* (IAR) [17] est plus prudente. Elle interroge une seule sous-base cohérente de la base ABox obtenue à partir de l’intersection de toutes les réparations. L’avantage de la réparation IAR est qu’elle peut être calculée en temps polynomial.

La notion de réparation non-contestée (en anglais *non-defeated*) d’une base ABox incohérente et totalement préordonnée a été proposée dans [4] dans le cadre des logiques DL-Lite et dans [6] dans le cadre de la logique propositionnelle. L’approche suppose que la base ABox est stratifiée du fait de l’application d’un préordre total sur les assertions. Intuitivement, la réparation non-contestée est basée sur l’application itérative de la sémantique IAR à des ensembles formés par un nombre de strates incrémenté à chaque itération. Ce calcul est également effectué en temps polynomial [21] pour DL-Lite.

Dans cet article, nous nous intéressons à la recherche d’un moyen efficace pour calculer les réparations d’une base de connaissances incohérente spécifiée en DL-Lite, et dans laquelle la relation de priorité sur les assertions est un préordre partiel. Pour ce faire, nous proposons une méthode efficace et saine, appelée Elect, pour restaurer la cohérence de la base ABox par rapport à la base TBox. Nous montrons que la méthode Elect généralise à la fois la sémantique IAR lorsque les bases assertionnelles sont non-ordonnées et aussi la sémantique “non-contestée” pour des bases ABox totalement préordonnées.

La justification sémantique de la méthode Elect consiste d’abord à interpréter un préordre partiel associé à une base ABox comme une famille de préordres totaux, puis d’appliquer dans un deuxième temps la sémantique “non-contestée” à chacune des bases ABox totalement préordonnées, et finalement de calculer leur intersection pour produire une réparation unique. La méthode Elect est saine puisqu’elle ne fait pas de choix arbitraire entre les préordres totaux associés à un préordre partiel. De ce fait, tous les préordres totaux sont pris en compte pour définir la méthode Elect. Nous proposons le concept d’assertion élue comme étant une assertion strictement préférée à toute autre assertion avec laquelle elle est en conflit. Cela nous permet de proposer une caractérisation équivalente de la méthode Elect, et donc d’obtenir la réparation sans avoir à calculer explicitement l’ensemble des préordres totaux. Enfin, nous montrons que le calcul de la réparation avec la méthode Elect se fait en temps polynomial. Ainsi, Elect préserve la calculabilité de la sémantique IAR et de la sémantique “non-contestée” pour des bases ABox partielle-

ment préordonnées.

Cet article est structuré comme suit. La section 2 contient des rappels sur la logique DL-Lite. La section 3 présente la sémantique IAR pour des bases ABox non-ordonnées. La section 4 aborde la réparation non-contestée pour des bases ABox totalement préordonnées. La section 5 concerne les bases ABox partiellement préordonnées. Nous introduisons notre méthode appelée Elect et en proposons une caractérisation. La section 6 contient une discussion sur trois extensions possibles de la méthode Elect, avant de conclure l’article dans la section 7.

## 2 La Logique de Description DL-Lite

Les Logiques de Description (DL) [1] sont une famille de formalismes de représentation des connaissances ayant de nombreuses applications, notamment pour la formalisation d’ontologies. Les fragments légers de DL tels que DL-Lite [12] sont particulièrement intéressants car ils offrent un bon compromis entre le pouvoir expressif et la complexité calculatoire. En effet, répondre à des requêtes en DL-Lite s’effectue de manière efficace. Il existe quelques variantes de DL-Lite, telles que DL-Lite<sub>R</sub> que nous utilisons dans le présent article.

Le langage DL-Lite<sub>R</sub> est construit sur un ensemble fini de *noms de concepts*  $C$ , un ensemble fini de *noms de rôles*  $R$  et un ensemble fini de *noms d’individus*  $I$ , tels que  $C$ ,  $R$  et  $I$  sont mutuellement disjoints. Le langage DL-Lite<sub>R</sub> est défini selon les règles suivantes :

$$\begin{aligned} R &\longrightarrow P \mid P^- & E &\longrightarrow R \mid \neg R \\ B &\longrightarrow A \mid \exists R & C &\longrightarrow B \mid \neg B \end{aligned}$$

Dans ces règles,  $A \in C$  dénote un nom de concept,  $P \in R$  dénote un nom de rôle, et  $P^-$  est la relation *inverse* associée à  $P$ . De plus,  $R$  dénote un *rôle de base*, alors que  $E$  représente un *rôle complexe*. Par ailleurs,  $B$  dénote un *concept de base* et  $C$  est un *concept complexe*.

**Exemple 1** *Considérons l’exemple suivant :*

–  $C = \{\text{Danse}, \text{Mdanse}, \text{Tdanse}, \text{DanseA}, \text{DanseS}, \text{Acc}\}$ , représentant respectivement les noms de concept : *danse, danse moderne, danse traditionnelle, danse avec des accessoires, danse sans accessoires ainsi que les accessoires utilisés dans certaines danses.*

–  $R = \{UtA\}$ , représentant les accessoires utilisés dans certaines danses et qui peuvent être des fleurs (fl en abrégé), un chapeau (ch en abrégé), ou des écharpes (ec en abrégé).

–  $I = \{d_1, d_2, d_3, d_4, d_5\} \cup \{fl, ch, ec\}$ , où chaque  $d_i$ , pour  $i = 1, \dots, 5$ , représente une danse et le reste représente des accessoires.

Des exemples de concepts complexes sont :  $\neg \text{DanseA}$  et  $\neg \exists UtA$ .  $\square$

Un *axiome d’inclusion* sur des concepts (respectivement des rôles) est un énoncé de la forme  $B \sqsubseteq C$  (respectivement,  $R \sqsubseteq E$ ). Une inclusion de concept ayant le symbole  $\neg$  du

côté droit de l'inclusion est appelée *axiome négatif d'inclusion*, sinon elle est appelée *axiome positif d'inclusion*. Des exemples d'axiomes d'inclusion sur des concepts sont :  $DanseS \sqsubseteq \neg DanseA$  et  $\exists UtA^- \sqsubseteq Acc$ .

Une base  $TBox \mathcal{T}$  en  $DL\text{-Lite}_R$  est un ensemble fini d'axiomes d'inclusion (positifs et négatifs). Une *assertion* est un énoncé de la forme  $A(a)$  ou  $P(a, b)$ , avec  $a, b \in I$ . Des exemples d'assertions sont  $Mdanse(d_1)$  et  $UtA(d_3, ch)$ . Une base  $ABox \mathcal{A}$  en  $DL\text{-Lite}_R$  est un ensemble fini d'assertions. Pour des bases  $\mathcal{T}$  et  $\mathcal{A}$ , nous notons une *base de connaissances* (KB) par  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ .

Dans cet article, nous utilisons l'exemple suivant.

**Exemple 1** (suite) Soit la base  $TBox$  suivante :

$$\mathcal{T} = \left\{ \begin{array}{l} 1. Mdanse \sqsubseteq Danse, \\ 2. Tdanse \sqsubseteq Danse, \\ 3. Tdanse \sqsubseteq DanseA, \\ 4. Mdanse \sqsubseteq DanseS, \\ 5. DanseS \sqsubseteq \neg DanseA, \\ 6. DanseS \sqsubseteq \neg \exists UtA, \\ 7. \exists UtA^- \sqsubseteq Acc, \\ 8. \exists UtA \sqsubseteq DanseA \end{array} \right\}$$

Les deux premiers axiomes indiquent seulement que les danses modernes et les danses traditionnelles sont des danses. L'axiome 3 signifie que les danses traditionnelles sont des danses qui utilisent des accessoires. L'axiome 4 énonce que les danses modernes n'utilisent pas d'accessoires. L'axiome 5 indique que l'ensemble de danses modernes et l'ensemble de danses traditionnelles sont disjoints. L'axiome 6 énonce qu'une danse moderne n'utilise pas d'accessoires. L'axiome 7 indique que les éléments utilisés par les danses, donnés par le nom de rôle "UtA", doivent appartenir à l'ensemble des éléments spécifiés par le nom de concept "Acc". L'axiome 8 spécifie que tout ce qui utilise des accessoires doit être une danse avec accessoires.

Soit la base  $ABox$  donnée par les assertions suivantes :

$$\mathcal{A} = \left\{ \begin{array}{l} Mdanse(d_1), Mdanse(d_2), Tdanse(d_2), \\ Tdanse(d_3), Tdanse(d_4), DanseA(d_3), \\ DanseA(d_5), DanseS(d_5), UtA(d_2, fl), \\ UtA(d_3, ch), UtA(d_4, ec) \end{array} \right\}$$

□

Une base de connaissances  $\mathcal{K}$  est dite *cohérente* si elle admet au moins un modèle, sinon elle est *incohérente*. Une base  $TBox \mathcal{T}$  est *incohérente* s'il existe un nom de concept  $A \in \mathcal{C}$  tel que  $A$  est vide dans chaque modèle de  $\mathcal{T}$ , sinon elle est *cohérente*. Notons que la base KB donnée par  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  dans l'Exemple 1 est incohérente.

Pour de plus amples détails sur la famille  $DL\text{-Lite}$  de Logiques de Description, nous invitons le lecteur à consulter les travaux de Calvanese et al. [12]. Dans la suite de cet article, nous notons  $DL\text{-Lite}_R$  simplement par  $DL\text{-Lite}$ .

### 3 Sémantique IAR pour des Bases Assertionnelles Non-Ordonnées

Dans cette section, nous considérons une base KB spécifiée en  $DL\text{-Lite}$   $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  qui peut être incohérente. Nous supposons que la base  $TBox \mathcal{T}$  est cohérente et fiable (c-à-d. validée par les concepteurs de l'ontologie). De ce fait, les éléments de  $\mathcal{T}$  ne sont pas remis en cause en présence de conflits, contrairement aux assertions de  $\mathcal{A}$  qui peuvent être discutables. De plus, nous supposons que la base  $ABox \mathcal{A}$  est non-ordonnée, donc toutes les assertions ont le même degré de priorité. Une manière standard de gérer l'incohérence consiste à d'abord calculer l'ensemble des sous-ensembles maximaux cohérents de  $\mathcal{A}$ , appelés réparations, puis de les utiliser pour l'inférence (c'est-à-dire pour répondre à des requêtes). Formellement, une réparation est définie comme suit [17] :

**Définition 1** Soit  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base KB non-ordonnée et incohérente. Une sous-base  $\mathcal{R} \subseteq \mathcal{A}$  est une réparation si  $\langle \mathcal{T}, \mathcal{R} \rangle$  est cohérente, et  $\forall \mathcal{R}' \subseteq \mathcal{A}$  t.q.  $\mathcal{R} \subsetneq \mathcal{R}'$ ,  $\langle \mathcal{T}, \mathcal{R}' \rangle$  est incohérente. De plus, si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est cohérente, alors il existe une seule réparation  $\mathcal{R} = \mathcal{A}$ .

De ce fait, lorsque  $\mathcal{K}$  est incohérente, ajouter une assertion  $f$  de  $(\mathcal{A} \setminus \mathcal{R})$  à  $\mathcal{R}$  entraîne l'incohérence de  $\langle \mathcal{T}, \mathcal{R} \cup \{f\} \rangle$ .

Nous notons par  $MAR(\mathcal{A})$  (de l'anglais *Maximal Assertional-based Repair*) l'ensemble des réparations de  $\mathcal{A}$  par rapport à  $\mathcal{T}$ . Grâce à la notion de réparation, l'incohérence d'une base KB non-ordonnée peut être gérée par l'application d'un mécanisme standard de réponse aux requêtes, soit à l'ensemble des réparations (conséquence universelle ou conséquence AR [17]), soit à une seule réparation (conséquence "brave" [10]). Il est bien connu que la sémantique dite brave est aventureuse et peut entraîner des conclusions contestables, tandis que la sémantique AR est saine mais coûteuse sur le plan calculatoire.

Une alternative à cela est d'utiliser la sémantique IAR [17] qui sélectionne une seule sous-base cohérente de  $\mathcal{A}$ , dénotée  $IAR(\mathcal{A})$ . Avant d'introduire la sémantique IAR, nous présentons d'abord la notion de conflit assertionnel. Il s'agit d'un sous-ensemble d'assertions minimal incohérent qui contredit la base  $TBox$ . Formellement :

**Définition 2** Soit  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base KB incohérente. Une sous-base  $C \subseteq \mathcal{A}$  est un *conflit assertionnel* dans  $\mathcal{K}$  ssi  $\langle \mathcal{T}, C \rangle$  est incohérente et  $\forall f \in C$ ,  $\langle \mathcal{T}, C \setminus \{f\} \rangle$  est cohérente.

Nous notons par  $C(\mathcal{A})$  l'ensemble des conflits de  $\mathcal{A}$ . D'après la Définition 2, nous constatons qu'enlever une assertion  $f$  de  $C$  restaure la cohérence de  $\langle \mathcal{T}, C \rangle$ . Une propriété intéressante de  $DL\text{-Lite}$  est que le calcul de l'ensemble des conflits se fait en temps polynomial [13]. Par ailleurs, un conflit  $C$  concerne exactement deux assertions [13]. Dans ce cas, si  $f$  et  $g$  sont deux assertions qui

appartiennent à un conflit, nous notons le conflit par une paire  $\{f, g\}$  et nous disons que  $f$  et  $g$  sont en conflit.

Nous introduisons à présent la notion d'éléments libres.

**Définition 3** Soit  $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$  une base KB incohérente. Une assertion  $f \in \mathcal{A}$  est dite libre ssi  $\forall C \in \mathcal{C}(\mathcal{A}) : f \notin C$ .

Intuitivement, les assertions libres correspondent aux éléments qui n'apparaissent dans aucun conflit. A l'origine, la notion d'éléments libres avait été proposée dans le cadre de la logique propositionnelle [5].

Dans la suite de cet article, nous notons par  $IAR(\mathcal{A})$  (de l'anglais *Intersection ABox Repair*) l'ensemble des éléments libres de  $\mathcal{A}$ . Formellement :

**Définition 4**  $IAR(\mathcal{A}) = \{f : f \in \mathcal{A} \text{ t.q. } f \text{ est libre}\}$ .

La Définition 4 est une réécriture équivalente de la définition standard de  $IAR(\mathcal{A})$  donnée par l'intersection de toutes les réparations :  $IAR(\mathcal{A}) = \bigcap \{\mathcal{R} \mid \mathcal{R} \in MAR(\mathcal{A})\}$  [5, 17].

Répondre à des requêtes avec la sémantique IAR revient à appliquer un mécanisme standard de réponse aux requêtes à la base  $\langle \mathcal{T}, IAR(\mathcal{A}) \rangle$  (du fait que  $\langle \mathcal{T}, IAR(\mathcal{A}) \rangle$  est cohérente).

**Exemple 2** L'ensemble des conflits dans  $\langle \mathcal{T}, \mathcal{A} \rangle$  est :

$$C(\mathcal{A}) = \left\{ \begin{array}{l} \{Mdanse(d_2), Tdanse(d_2)\}, \\ \{Mdanse(d_2), UtA(d_2, fl)\}, \\ \{DanseA(d_5), DanseS(d_5)\} \end{array} \right\}$$

Afin d'obtenir  $IAR(\mathcal{A})$ , il suffit d'enlever de  $\mathcal{A}$  toutes les assertions de  $C(\mathcal{A})$ . Cela donne :

$$IAR(\mathcal{A}) = \left\{ \begin{array}{l} Mdanse(d_1), Tdanse(d_3), Tdanse(d_4), \\ DanseA(d_3), UtA(d_3, ch), UtA(d_4, ec) \end{array} \right\}$$

## 4 Réparation Non-Contestée pour des Bases Assertionnelles Totalemt Préordonnées

Dans cette section, nous considérons des bases KB spécifiées en DL-Lite, incohérentes et totalemt préordonnées. Donc une relation de préordre total  $\geq$  est appliquée seulement à la composante ABox que nous notons par  $(\mathcal{A}, \geq)$ . La relation  $\geq$  est réflexive, transitive et vérifie :  $\forall f \in \mathcal{A}, \forall g \in \mathcal{A}$ , soit  $f \geq g$  ou bien  $g \geq f$ . Soient  $>$  la relation de préférence stricte et  $\equiv$  la relation d'équivalence associées à la relation  $\geq$ . De plus, nous représentons  $(\mathcal{A}, \geq)$  par la partition bien-ordonnée de  $\mathcal{A}$  induite par  $\geq$ . Ainsi, étant donné  $(\mathcal{A}, \geq)$ , nous considérons que  $\mathcal{A}$  est partitionnée en  $n$  strates de la forme  $\mathcal{A} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$ , tel que :

- $\mathcal{S}_1 = \{f \in \mathcal{A} : \forall g \in \mathcal{A}, f \geq g\}$ , et
- $\mathcal{S}_i = \{f \in \mathcal{A} : \forall g \in \mathcal{A} \setminus (\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i-1}), f \geq g\}$ , pour  $i = 2, \dots, n$ .

En d'autres termes, les assertions d'une strate  $\mathcal{S}_i$  ont le même degré de priorité  $i$ , et elles sont plus fiables que celles

d'une strate  $\mathcal{S}_j$  pour  $j > i$ . Donc  $\mathcal{S}_1$  contient les assertions les plus importantes, alors que  $\mathcal{S}_n$  contient les assertions les moins importantes, et  $\mathcal{A} = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_n$ .

Plusieurs travaux prennent en compte la notion de priorité pour interroger des bases de données incohérentes (comme [18, 20]) ou des bases de connaissances DL (comme [9, 15]). La plupart de ces formalismes se basent sur les notions de réparation et de sémantique AR, ils sont donc coûteux sur le plan calculatoire. En particulier, la sémantique "des réparations préférées" a été introduite dans [9] (dans l'esprit de ce qui avait été proposé en logique propositionnelle pondérée [19, 11]). Cette sémantique revisite les sémantiques AR et IAR en remplaçant la notion de réparation par différents types de réparations préférées basées sur : le cardinal d'un ensemble, des degrés de priorité sur la base ABox et des poids sur les assertions. Cependant, ce formalisme entraîne souvent une augmentation de la complexité pour les sémantiques proposées. Notamment, la complexité de la sémantique IAR qui est polynomiale dans un cadre sans priorités explose lorsqu'un préordre total est appliqué à la base ABox.

Dans [4], une attention particulière a été accordée aux approches qui sélectionnent une seule réparation. L'une de ces approches est la réparation non-contestée qui a une complexité polynomiale sans être aventureuse (c'est-à-dire qui génère des conclusions contestables). Intuitivement, la réparation non-contestée consiste à récupérer itérativement, strate par strate, l'ensemble des assertions libres.

**Définition 5** Soit  $\mathcal{K}$  une base KB incohérente dont la base ABox  $(\mathcal{A}, \geq)$  est totalemt préordonnée. Soit  $\mathcal{A} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$  la partition bien-ordonnée associée à  $\geq$ .

La réparation non-contestée  $nd(\mathcal{A}, \geq) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n$  t.q. :

$$\forall i = 1, \dots, n : \mathcal{S}'_i = IAR(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)$$

où  $\forall i : IAR(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)$  est la base IAR de  $(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)$ , donnée par la Définition 4.

La définition d'une sous-base non-contestée est une adaptation de celle proposée dans le cadre de la logique propositionnelle [6]. Cependant, la réparation non-contestée est calculée en temps polynomiale dans le cadre de DL-Lite, alors que son calcul est difficile en logique propositionnelle.

Notons qu'une réécriture (similaire à celle de  $IAR(\mathcal{A})$ ) est donnée pour  $nd(\mathcal{A}, \geq)$  dans [3]. En fait, une assertion  $f \in \mathcal{S}_i$  est dite contestée s'il existe une assertion  $g \in \mathcal{S}_j$  telle que  $j \leq i$ , et  $\{f, g\}$  est un conflit. Il a été démontré que  $nd(\mathcal{A}, \geq)$  contient toutes les assertions non-contestées [3].

**Exemple 3** Reprenons notre exemple et considérons un préordre total  $\geq$  sur les assertions de la base ABox représenté par la Figure 1, où :

- $f \equiv g$  signifie que les deux assertions ont le même degré de priorité,
- la flèche  $f \rightarrow g$  signifie que  $f$  est plus prioritaire que  $g$  (c-à-d.  $f > g$ ).

$$\begin{aligned}
& Tdanse(d_2) \equiv Mdanse(d_1) \equiv Tdanse(d_3) \equiv UtA(d_3, ch) \\
& \quad \downarrow \\
& UtA(d_2, fl) \equiv DanseA(d_5) \equiv DanseS(d_5) \equiv DanseA(d_3) \\
& \quad \downarrow \\
& Mdanse(d_2) \equiv Tdanse(d_4) \equiv UtA(d_4, ec)
\end{aligned}$$

FIGURE 1 – Un préordre total sur la base ABox

A partir de cette ABox totalement préordonnée, il est possible de calculer la sous-classe non-contestée de  $\mathcal{A}$ . La partition bien-ordonnée correspondante est :

- $\mathcal{S}_1 = \{Mdanse(d_1), Tdanse(d_2), Tdanse(d_3), UtA(d_3, ch)\}$ .
- $\mathcal{S}_2 = \{UtA(d_2, fl), DanseA(d_3), DanseA(d_5), DanseS(d_5)\}$ .
- $\mathcal{S}_3 = \{Mdanse(d_2), Tdanse(d_4), UtA(d_4, ec)\}$ .

Nous avons :

$$nd(\mathcal{A}, \geq) = IAR(\mathcal{S}_1) \cup IAR(\mathcal{S}_1 \cup \mathcal{S}_2) \cup IAR(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3),$$

tel que :

- $IAR(\mathcal{S}_1) = \{Tdanse(d_2), Mdanse(d_1), Tdanse(d_3), UtA(d_3, ch)\}$ .
- $IAR(\mathcal{S}_1 \cup \mathcal{S}_2) = \{Tdanse(d_2), Mdanse(d_1), Tdanse(d_3), UtA(d_3, ch), UtA(d_2, fl), DanseA(d_3)\}$ .
- $IAR(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3) = IAR(\mathcal{A})$  (donné dans l'exemple 2).

Par conséquent :

$$nd(\mathcal{A}, \geq) = \{Mdanse(d_1), Tdanse(d_2), Tdanse(d_3), Tdanse(d_4), DanseA(d_3), UtA(d_2, fl), UtA(d_3, ch), UtA(d_4, ec)\}.$$

## 5 Bases Assertionnelles Partiellement Préordonnées

La sémantique IAR (ABox non-ordonnée) et la sémantique “non-contestée” (ABox totalement préordonnée) sont connues pour leur efficacité pour gérer l'incohérence. En effet, elles calculent une réparation unique pour la base ABox et le font en temps polynomial. Dans cette section, nous proposons une méthode pour calculer une réparation unique lorsqu'un préordre partiel noté  $\trianglerighteq$  est appliqué aux assertions de la base ABox notée  $(\mathcal{A}, \trianglerighteq)$ . Soient  $\triangleright$  l'ordre strict (irréflexif et transitif) et  $\equiv$  la relation d'équivalence associés à la relation  $\trianglerighteq$ .

Nous appelons notre méthode “Elect” et notons par  $Elect(\mathcal{A}, \trianglerighteq)$  la réparation qu'elle produit. Comme nous le verrons par la suite, la méthode Elect généralise la sémantique IAR dans le cas où la relation  $\trianglerighteq$  est non-ordonnée, et aussi la sémantique “non-contestée” lorsque la relation  $\trianglerighteq$  est totalement préordonnée.

### 5.1 Une famille de préordres totaux à partir d'un préordre partiel

Dans cette section, nous considérons qu'un préordre partiel  $\trianglerighteq$  est interprété comme une famille de préordres totaux,

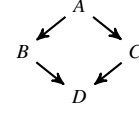


FIGURE 2 – Un préordre partiel sur la base ABox

de sorte que chaque préordre total est une extension totale de la relation  $\trianglerighteq$  définie comme suit :

**Définition 6** Un préordre total  $\geq$  sur  $\mathcal{A}$  est une extension totale de  $\trianglerighteq$  sur  $\mathcal{A}$  ssi  $\forall f, g \in \mathcal{A}$ , si  $f \trianglerighteq g$  alors  $f \geq g$ .

Notons que l'interprétation d'une base KB partiellement préordonnée comme une famille de bases KB totalement préordonnées est une représentation naturelle qui a été utilisée dans d'autres cadres tels que la logique possibiliste partiellement ordonnée [7, 22] et les réseaux probabilistes crédaux [14]. Cette démarche n'a de sens que si les éléments peuvent être comparés (le fait de dire que deux assertions  $f$  et  $g$  sont incomparables signifie que soit  $f$  est strictement préférée à  $g$ , soit  $f$  est aussi préférée que  $g$ , soit  $g$  est strictement préférée à  $f$ , mais on ignore lequel).

**Exemple 4** Soit un préordre partiel  $\trianglerighteq$  sur les assertions de la base ABox qui permet de les répartir en quatre sous-ensembles comme suit :

- $A = \{Mdanse(d_1) \equiv Tdanse(d_2) \equiv Tdanse(d_3) \equiv UtA(d_3, ch)\}$
- $B = \{UtA(d_2, fl) \equiv DanseA(d_3) \equiv DanseS(d_5) \equiv DanseA(d_5)\}$
- $C = \{Mdanse(d_2)\}$
- $D = \{Tdanse(d_4) \equiv UtA(d_4, ec)\}$

où  $f \equiv g$  signifie que les assertions  $f$  et  $g$  ont le même degré de priorité.

Ce préordre partiel est représenté par la Figure 2, où la flèche  $A \rightarrow B$  (par exemple) signifie  $\forall f \in A, \forall g \in B$ , l'assertion  $f$  est plus prioritaire que  $g$  (c-à-d.  $f \triangleright g$ ).

De ce fait, l'ensemble  $A$  (resp.  $D$ ) contient les assertions les plus (resp. les moins) prioritaires. Les assertions des ensembles  $B$  et  $C$  sont incomparables.

Il s'en suit que le préordre partiel  $\trianglerighteq$  peut être interprété comme une famille de trois préordres totaux  $\geq_1, \geq_2$  et  $\geq_3$  représentés par la Figure 3, de sorte que :

- Selon  $\geq_1$ , les assertions de l'ensemble  $B$  sont strictement préférées à celles de l'ensemble  $C$ .
- Selon  $\geq_2$ , les assertions des ensembles  $B$  et  $C$  sont préférées de manière égale.
- Selon  $\geq_3$ , les assertions de l'ensemble  $C$  sont strictement préférées à celles de l'ensemble  $B$ .  $\square$

Une question qui se pose est comment gérer cette famille de ABox totalement préordonnées. Nous voulons éviter le

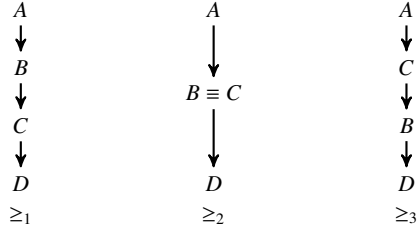


FIGURE 3 – Les extensions totales du préordre partiel  $\succeq$

choix arbitraire qui consiste à sélectionner un seul préordre total parmi tous les autres. Il faudrait donc prendre en compte tous les préordres totaux. Une méthode prudente pour obtenir une sous-base assertionnelle cohérente unique est de considérer l'intersection de toutes les réparations non-contestées associées à tous les préordres totaux.

**Définition 7** Soit  $\mathcal{K}$  une base KB spécifiée en DL-Lite, incohérente et dont la base ABox  $(\mathcal{A}, \succeq)$  est partiellement préordonnée.

–  $Elect(\mathcal{A}, \succeq) = \bigcap_{\succeq} \{nd(\mathcal{A}, \succeq) \text{ t.q. } \succeq \text{ est une extension totale de } \succeq\}$ , où  $nd(\mathcal{A}, \succeq)$  est donné par la Définition 5.

– Soit  $q$  une requête. Alors  $q$  est une conséquence de  $\mathcal{K}$  par la méthode *Elect* si  $q$  s'ensuit de  $Elect(\mathcal{A}, \succeq)$  (en utilisant l'inférence standard de DL-lite).

Nous illustrons cette idée sur notre exemple.

**Exemple 5** Les réparations non-contestées des ABox totalement préordonnées  $(\mathcal{A}, \succeq_1)$ ,  $(\mathcal{A}, \succeq_2)$  et  $(\mathcal{A}, \succeq_3)$  sont :

–  $nd(\mathcal{A}, \succeq_1) = A \cup \{UtA(d_2, fl), DanseA(d_3)\} \cup D$ .

–  $nd(\mathcal{A}, \succeq_2) = A \cup \{DanseA(d_3)\} \cup D$ .

–  $nd(\mathcal{A}, \succeq_3) = A \cup \{DanseA(d_3)\} \cup D$ .

Par conséquent :

$Elect(\mathcal{A}, \succeq) = nd(\mathcal{A}, \succeq_1) \cap nd(\mathcal{A}, \succeq_2) \cap nd(\mathcal{A}, \succeq_3)$   
 $= A \cup \{DanseA(d_3)\} \cup D$ .  $\square$

Dans ce qui suit, nous étudions quelques caractéristiques de notre méthode.

Un résultat important énoncé dans la Proposition 1 est que le calcul de la réparation  $Elect(\mathcal{A}, \succeq)$  peut s'effectuer en temps polynomial. En effet, il n'est pas nécessaire d'exhiber toutes les extensions totales de la relation  $\succeq$ .

**Proposition 1** Le calcul de la réparation  $Elect(\mathcal{A}, \succeq)$  se fait en temps polynomial (par rapport à la taille de  $\mathcal{A}$ ).

La proposition suivante établit qu'une base KB ayant  $Elect(\mathcal{A}, \succeq)$  comme base ABox est cohérente.

**Proposition 2**  $\langle \mathcal{T}, Elect(\mathcal{A}, \succeq) \rangle$  est cohérente.

Une autre caractéristique de la méthode *Elect* est qu'elle correspond à la sémantique IAR lorsque la base ABox est non-ordonnée, et à la sémantique "non-contestée" lorsque la base ABox est totalement préordonnée.

### Proposition 3

– Si le préordre partiel  $\succeq$  est non-ordonné, alors  $Elect(\mathcal{A}, \succeq) = IAR(\mathcal{A})$ .

– Si le préordre partiel  $\succeq$  est un préordre total, alors  $Elect(\mathcal{A}, \succeq) = nd(\mathcal{A}, \succeq)$ .

Les preuves des Propositions 1 à 3 sont établies grâce à la définition d'une caractérisation équivalente de la réparation  $Elect(\mathcal{A}, \succeq)$ , présentée dans la prochaine section.

## 5.2 Caractérisation de la réparation $Elect(\mathcal{A}, \succeq)$

Dans cette section, nous proposons une caractérisation de la réparation  $Elect(\mathcal{A}, \succeq)$  qui permet d'éviter de calculer l'ensemble des extensions totales de la relation  $\succeq$ . Pour cela, nous introduisons le concept d'assertions élues. Intuitivement, une assertion est élue dans  $(\mathcal{A}, \succeq)$  si elle est strictement préférée à toutes les assertions avec lesquelles elle est en conflit. Formellement :

**Définition 8** Une assertion  $f \in \mathcal{A}$  est élue ssi  $\forall g \in \mathcal{A}$ , si  $\{f, g\}$  est un conflit, alors  $f \triangleright g$  (c-à-d.  $f$  est strictement préférée à  $g$ ).

La Définition 8 généralise le concept d'assertions libres donné dans la Définition 3. En effet, si la relation  $\succeq$  est non-prioritaire (c-à-d.  $\forall f \in \mathcal{A}, \forall g \in \mathcal{A}, f \succeq g$  et  $g \succeq f$ ), alors  $f$  est élue dans  $(\mathcal{A}, \succeq)$  ssi  $f$  est libre. Evidemment, le contraire n'est en général pas vrai (lorsque la relation  $\succeq$  est prioritaire). En effet, une assertion élue peut ne pas être une assertion libre, mais son degré de fiabilité est strictement plus important que celui de ses opposants. Cette définition généralise également la notion d'assertions acceptées donnée pour des réparations non-contestées dans des bases KB totalement préordonnées [3]. Enfin, le concept d'assertions élues est dans le même esprit que la notion de croyances acceptées proposée en théorie de l'incertitude [16].

La Proposition 4 énonce que les assertions élues sont exactement celles de la réparation  $Elect(\mathcal{A}, \succeq)$ .

**Proposition 4** Une assertion  $f \in \mathcal{A}$  est élue dans  $(\mathcal{A}, \succeq)$  ssi  $f \in Elect(\mathcal{A}, \succeq)$ .

Nous démontrons cette proposition comme suit.

**Preuve 1** Soit  $(\mathcal{A}, \succeq)$  une base assertionnelle partiellement préordonnée.

1. Soit  $f \in \mathcal{A}$  une assertion élue. Montrons que pour chaque extension totale  $(\mathcal{A}, \succeq)$  de  $(\mathcal{A}, \succeq)$ , nous avons  $f \in nd(\mathcal{A}, \succeq)$ . Soit  $(S_1, \dots, S_n)$  la partition bien-ordonnée associée à  $\succeq$ . Soit  $i$  la première strate où  $f \in S_i$ .

Rappelons que  $f$  est élue dans  $(\mathcal{A}, \succeq)$  signifie  $\forall g \in \mathcal{A}$ , si  $\{f, g\}$  est un conflit, alors  $f \triangleright g$  (c-à-d.  $f$  est strictement préférée à  $g$  selon  $\succeq$ ). Et puisque  $\succeq$  est une extension totale de  $\succeq$ , alors cela signifie que  $f > g$ . Cela signifie aussi  $\forall g \in \mathcal{A}$  tel que  $\{f, g\}$  est un conflit,  $g \in S_j$  avec  $j > i$ . Donc,  $f \in IAR(S_1 \cup \dots \cup S_i)$ . Il s'en suit que  $f \in nd(\mathcal{A}, \succeq)$ .

2. Montrons à présent l'inverse. Supposons que  $f \in \mathcal{A}$  n'est pas élue et construisons une extension totale  $(\mathcal{A}, \geq)$  de  $(\mathcal{A}, \triangleright)$  telle que  $f \notin nd(\mathcal{A}, \geq)$ .

L'assertion  $f$  n'est pas élue signifie  $\exists g \in \mathcal{A}$  tel que  $\{f, g\}$  est un conflit et  $f \triangleright g$  n'est pas vrai. Donc il existe une extension totale  $\geq$  de  $\triangleright$  où  $g \geq f$ . Si  $\{f, g\}$  est un conflit et  $(S_1, \dots, S_n)$  est la partition bien-ordonnée associée à  $\geq$ , alors si  $f \in S_i$ , il s'en suit que  $g \in S_j$  où  $j \leq i$ . Par conséquent,  $\forall k \in \{1, \dots, n\}$ ,  $f \notin IAR(S_1 \cup \dots \cup S_k)$ , donc  $f \notin nd(\mathcal{A}, \geq)$ .  $\square$

A partir de ce résultat, nous pouvons démontrer les Propositions 1, 2 et 3 données dans la section précédente.

### Preuve 2

1. En ce qui concerne la complexité calculatoire, nous rappelons que le calcul de l'ensemble des conflits  $C(\mathcal{A})$  se fait en temps polynomial par rapport à la taille de  $\mathcal{A}$ . Donc, calculer  $Elect(\mathcal{A}, \geq)$  se fait aussi en temps polynomial. En effet, vérifier si une assertion  $f \in \mathcal{A}$  est élue revient à parcourir tous les conflits assertionnels dans  $C(\mathcal{A})$ . Ceci se fait en temps linéaire par rapport à la taille de  $C(\mathcal{A})$  (cette taille étant bornée par  $O(|\mathcal{A}|^2)$ ).

2. Montrons que la base  $Elect(\mathcal{A}, \geq)$  est cohérente par rapport à  $\mathcal{T}$ . Supposons que ce n'est pas le cas. Donc  $\exists f \in Elect(\mathcal{A}, \geq)$ ,  $\exists g \in Elect(\mathcal{A}, \geq)$ ,  $g \neq f$ , tel que  $\{f, g\}$  est un conflit. Puisque  $f$  et  $g$  sont dans  $Elect(\mathcal{A}, \geq)$ , alors cela signifie que  $f \triangleright g$  et  $g \triangleright f$ , ce qui est impossible.

3. Finalement, par construction de la base  $Elect(\mathcal{A}, \geq)$ , il est facile de vérifier que lorsque  $\geq$  est un préordre total, alors  $Elect(\mathcal{A}, \geq)$  est la réparation non-contestée de  $\triangleright$ . Et lorsque  $\geq$  est non-ordonnée (donc  $\forall f \in \mathcal{A}, \forall g \in \mathcal{A}, f \triangleright g$  et  $g \triangleright f$ ), alors  $Elect(\mathcal{A}, \geq) = IAR(\mathcal{A}) = \{f \in \mathcal{A} : \nexists g \in \mathcal{A}, t.q. \{f, g\} \text{ est un conflit}\}$ .  $\square$

**Exemple 6** Nous utilisons la notion d'assertions élues de la Définition 8 pour recalculer la réparation  $Elect(\mathcal{A}, \geq)$ . Il est aisé de vérifier que :

- L'assertion  $Mdanse(d_2)$  n'est pas élue car  $\{Mdanse(d_2), Tdanse(d_2)\}$  est un conflit et la préférence stricte  $Mdanse(d_2) \triangleright Tdanse(d_2)$  n'est pas vérifiée.
- Les assertions  $DanseS(d_5)$  et  $DanseA(d_5)$  ne sont pas élues car elles sont en conflit et ont le même degré de priorité.
- L'assertion  $UtA(d_2, fl)$  n'est pas élue car  $\{Mdanse(d_2), UtA(d_2, fl)\}$  est un conflit et la préférence stricte  $UtA(d_2, fl) \triangleright Mdanse(d_2)$  n'a pas lieu.
- Les assertions restantes sont toutes élues. A savoir :  $Elect(\mathcal{A}, \geq) = \{Mdanse(d_1), Tdanse(d_2), Tdanse(d_3), Tdanse(d_4), DanseA(d_3), UtA(d_3, ch), UtA(d_4, ec)\}$ .

Ce résultat correspond à la réparation calculée dans l'Exemple 5 en considérant toutes les extensions totales du préordre partiel  $\geq$ .

## 6 Extensions de la Méthode Elect

Dans cette section, nous proposons une discussion brève de trois extensions possibles de la méthode Elect. Leur étude approfondie fera l'objet de travaux futurs.

### 6.1 Au-delà de la méthode Elect

Une question qui se pose est comment obtenir une base plus grande (plus productive) qu'une base calculée par la méthode Elect, tout en produisant une réparation saine. Une solution immédiate est d'utiliser la notion de fermeture déductive positive, selon laquelle la fermeture de la base ABox est définie en termes des axiomes positifs de la base  $\mathcal{T}$ . Nous introduisons l'opérateur de fermeture [12, 4] comme suit :

**Définition 9** Soit  $\mathcal{T}_p$  l'ensemble de tous les axiomes d'inclusions positifs de  $\mathcal{T}$ . La fermeture déductive de  $\mathcal{A}$  par rapport à  $\mathcal{T}$  est :

$cl(\mathcal{A}) = \{B(a) : \langle \mathcal{T}_p, \mathcal{A} \rangle \models B(a) \text{ t.q. } B \text{ est un concept de } \mathcal{T} \text{ et } a \text{ est un individu de } \mathcal{A}\} \cup \{R(a, b) : \langle \mathcal{T}_p, \mathcal{A} \rangle \models R(a, b) \text{ t.q. } R \text{ est un rôle de } \mathcal{T}, \text{ et } a \text{ et } b \text{ sont des individus de } \mathcal{A}\}$ . Ici,  $\models$  est une relation d'inférence standard en DL-Lite.

Il y a deux façons d'appliquer la fermeture déductive positive, soit sur la base ABox initiale, soit sur l'ensemble des réparations non-contestées calculées pour toutes les extensions totales du préordre partiel.

Dans la première option, l'application de la fermeture positive à la base ABox initiale, dans l'esprit de la sémantique ICAR pour des bases ABox non-ordonnées [17], soulève deux questions. Premièrement, en termes de sémantique, l'approche ICAR peut ne pas être fiable puisqu'elle permet de dériver des conséquences à partir d'assertions discutables. Deuxièmement, il peut y avoir plusieurs façons de définir la fiabilité des éléments dérivés. Par exemple, supposons que la base TBox contient  $\{A \sqsubseteq B, E \sqsubseteq B\}$  et que la base ABox contient  $\{A(x), E(x)\}$ . Supposons que  $A(x)$  et  $E(x)$  sont incomparables. Donc  $B(x)$  peut être dérivée de  $A(x)$  mais aussi de  $E(x)$ . La question est alors où positionner  $B(x)$ . L'intuition est de considérer  $B(x)$  comme *au moins aussi plausible que*  $A(x)$  et  $E(x)$ , mais cela n'est pas évident à définir de manière générale (en particulier pour des logiques de description expressives).

La deuxième option consiste à définir la fermeture de toutes les réparations non-contestées, ce qui est plus sain. En effet, les conclusions supplémentaires sont obtenues uniquement d'assertions appartenant à des réparations non-contestées, qui sont connues pour ne contenir que des assertions saines. Ainsi, la réparation associée à un préordre partiel est calculée comme l'intersection des réparations non-contestées fermées [4], une méthode que nous appelons CElect.



**Définition 10** Soit  $(\mathcal{A}, \triangleright)$  une base ABox partiellement préordonnée,  $cl(\cdot)$  un opérateur de fermeture donné par la Définition 9, et  $nd(\mathcal{A}, \geq)$  est donné par la Définition 5.

$CElect(\mathcal{A}, \triangleright) = \bigcap_{\geq} \{cl(nd(\mathcal{A}, \geq)) \text{ t.q. } \geq \text{ est une extension totale de } \triangleright\}$ .

Une base calculée avec la méthode CElect est alors plus grande qu’une base calculée avec la méthode Elect. De plus, pour des bases ABox non-ordonnées, une réparation calculée avec la méthode CElect est équivalente à la fermeture des réparations IAR (ce qui est différent d’une réparation calculée avec la méthode ICAR). Et pour des bases ABox totalement préordonnées, une réparation calculée avec la méthode CElect est équivalente à la fermeture des réparations non-contestées. Dans des travaux futurs, nous envisageons d’exhiber des cas spécifiques où la complexité de la méthode CElect serait également polynomiale en DL-Lite.

## 6.2 Au-delà de la réparation non-contestée

La question traitée dans cette partie concerne la possibilité ou non d’utiliser une sémantique autre que la sémantique “non-contestée” pour définir la méthode Elect. D’un point de vue sémantique, la réponse est positive. Par exemple, il est possible d’utiliser l’une des réparations préférées définies dans [9] au lieu de la réparation non-contestée dans notre définition de  $Elect(\mathcal{A}, \triangleright)$ . Rappelons d’abord la notion de réparation préférée pour des bases ABox totalement préordonnées.

**Définition 11** Soit  $\mathcal{A} = (S_1, \dots, S_n)$  une base ABox totalement préordonnée. Soient  $\mathcal{R}_1$  et  $\mathcal{R}_2$  deux sous-bases cohérentes de  $S_1 \cup \dots \cup S_n$ .

–  $\mathcal{R}_1$  est aussi préférée que  $\mathcal{R}_2$  ssi  $\forall i, 1 \leq i \leq n, \mathcal{R}_1 \cap S_i = \mathcal{R}_2 \cap S_i$ .

–  $\mathcal{R}_1$  est strictement préférée à  $\mathcal{R}_2$  ssi  $\exists i, 1 \leq i \leq n, \text{ t.q. } \mathcal{R}_2 \cap S_i \subsetneq \mathcal{R}_1 \cap S_i$  et  $\forall j, 1 \leq j < i, \mathcal{R}_1 \cap S_j = \mathcal{R}_2 \cap S_j$ .

Donc une base  $\mathcal{R} \subseteq \mathcal{A}$  est une réparation préférée ssi  $\nexists \mathcal{R}' \subseteq \mathcal{A} \text{ t.q. } \mathcal{R}' \text{ est strictement préférée à } \mathcal{R}$ .

(Se référer à [9] et aussi [6] pour de plus amples détails.)

La notion de réparation préférée définie pour un préordre total peut alors servir de base pour définir une réparation en présence d’un préordre partiel  $\triangleright$ .

Nous appelons ce nouveau formalisme  $Partial_{PR}(\mathcal{A}, \triangleright)$  (où  $PR$  désigne en anglais *Preferred Repairs*). Tout comme une réparation  $Elect(\mathcal{A}, \triangleright)$ , une réparation  $Partial_{PR}(\mathcal{A}, \triangleright)$  considère toutes les extensions totales  $\geq$  du préordre partiel  $\triangleright$ . Cependant, au lieu de considérer l’intersection des réparations non-contestées comme pour calculer  $Elect(\mathcal{A}, \triangleright)$ , nous considérons l’intersection des réparations préférées, notée  $IPR$  (de l’anglais *Intersection of Preferred Repairs*), comme suit :

**Définition 12** Soit  $(\mathcal{A}, \triangleright)$  une base ABox partiellement préordonnée. Soit  $\geq$  une extension totale de  $\triangleright$ , et  $(\mathcal{A}, \geq)$  la base ABox totalement préordonnée correspondante.

– L’intersection des réparations préférées (au sens de la Définition 11) associées à  $(\mathcal{A}, \geq)$  est :

$IPR(\mathcal{A}, \geq) = \bigcap \{\mathcal{R} \mid \mathcal{R} \text{ est une réparation préférée de } \geq\}$ .

– La réparation préférée associée à  $(\mathcal{A}, \triangleright)$  est :

$Partial_{PR}(\mathcal{A}, \triangleright) = \bigcap_{\geq} \{IPR(\mathcal{A}, \geq) \text{ t.q. } \geq \text{ est une extension totale de } \triangleright\}$ .

La réparation  $Partial_{PR}(\mathcal{A}, \triangleright)$  est plus grande que la réparation  $Elect(\mathcal{A}, \triangleright)$ . Cependant, la base  $Partial_{PR}(\mathcal{A}, \triangleright)$  n’est pas calculable en temps raisonnable puisque la complexité de  $IPR(\mathcal{A}, \triangleright)$  est coNP [9] si la relation  $\triangleright$  est un simple préordre total. Dans ce cas, la Proposition 1 n’est pas vérifiée.

## 6.3 Au-delà de DL-Lite

Une autre question concerne le fait de pouvoir ou non généraliser la méthode Elect à des bases ABox partiellement préordonnées exprimées dans des logiques autres que DL-Lite. D’un point de vue sémantique, il n’y a pas de limitations et les résultats obtenus correspondent également à la réparation IAR (base ABox non-ordonnée) et à la réparation non-contestée (base ABox totalement préordonnée). En effet, les sémantiques IAR et “non-contestée” sont définies indépendamment de la taille des conflits. En fait, la réparation IAR est simplement l’intersection de toutes les réparations maximales, et la réparation non-contestée est exprimée en termes de réparations IAR.

Cependant, d’un point de vue calculatoire, il est impératif d’avoir un moyen efficace de gérer les conflits afin que la Proposition 1 soit vérifiée. En particulier, il n’est pas nécessaire que les conflits assertionnels  $C \in C(\mathcal{A})$  soient binaires (c-à-d. impliquant deux assertions), à condition qu’ils puissent être calculés en temps polynomial.

Dans le cas de conflits impliquant plus de deux assertions, nous redéfinissons la notion d’assertion élue comme étant celle qui est strictement préférée à au moins une assertion de chaque conflit où elle apparaît. Nous utilisons le terme d’assertion élue  $y$  compris pour des conflits non-binaires.

**Définition 13** Soit  $\mathcal{K}$  une base KB spécifiée en Logique de Description, incohérente et dont la base ABox  $(\mathcal{A}, \triangleright)$  est partiellement préordonnée. Une assertion  $f \in \mathcal{A}$  est élue ssi  $\forall C \in C(\mathcal{A})$  où  $f \in C, \exists g \in C, g \neq f, \text{ tel que } f \triangleright g$  (c-à-d. l’assertion  $f$  est strictement préférée à  $g$ ).

Notons que lorsque  $C$  est un conflit binaire, la Définition 13 correspond à la Définition 8.

Pour des langages plus expressifs que DL-Lite, nous définissons formellement la réparation basée sur cette notion d’assertions élues. Nous la nommons “réparation dl-Elect”.

**Définition 14** Soit  $\mathcal{K}$  une base KB spécifiée en Logique de Description, incohérente et dont la base ABox  $(\mathcal{A}, \succeq)$  est partiellement préordonnée. Alors :

$dl\text{-Elect}(\mathcal{A}, \succeq) = \bigcap_{\succeq} \{nd(\mathcal{A}, \succeq) \text{ t.q. } \succeq \text{ est une extension totale de } \succeq\}$ , où  $nd(\mathcal{A}, \succeq)$  est donné par la Définition 5.

Nous proposons une caractérisation (sans avoir à spécifier tous les préordres totaux) de la réparation  $dl\text{-Elect}(\mathcal{A}, \succeq)$ .

**Proposition 5**

1. Une assertion  $f$  est élue dans  $(\mathcal{A}, \succeq)$  ssi  $f \in dl\text{-Elect}(\mathcal{A}, \succeq)$  (où une assertion élue est donnée par la Définition 13, et  $dl\text{-Elect}(\mathcal{A}, \succeq)$  est donné par la Définition 14).

2.  $dl\text{-Elect}(\mathcal{A}, \succeq)$  est cohérent par rapport à  $\mathcal{T}$ .

**Preuve 3** Soit  $(\mathcal{A}, \succeq)$  une base assertionnelle partiellement préordonnée.

1.i) Soit  $f \in \mathcal{A}$  une assertion élue. Montrons que pour chaque extension totale  $\succeq$  de  $\succeq$ , nous avons  $f \in nd(\mathcal{A}, \succeq)$ . Soit  $(S_1, \dots, S_n)$  la partition bien-ordonnée associée à  $\succeq$ . Supposons que  $f \in S_i$  pour un certain  $i \in \{1, \dots, n\}$ .

Puisque  $f$  est élue dans  $(\mathcal{A}, \succeq)$  et  $\succeq$  est une extension totale de  $\succeq$ , appliquer la Définition 13 entraîne  $\forall C \in C(\mathcal{A})$  où  $f \in C, \exists g \in C, g \neq f, \text{ t.q. } f \triangleright g$  et également  $f > g$ . Cela signifie aussi  $\forall C \in C(\mathcal{A}) \text{ t.q. } f \in C, \exists g \in C \text{ t.q. } g \neq f$  et  $g \in S_j$  avec  $j > i$ . Donc, il n'y a aucun conflit  $C$  dans  $S_1 \cup \dots \cup S_i$  t.q.  $f \in C$ . (Rappelons qu'un conflit est un ensemble d'assertions minimal incohérent par rapport à  $\mathcal{T}$ . Donc, enlever un élément de  $C$  produit un ensemble d'assertions cohérent par rapport à  $\mathcal{T}$ .)

Ainsi,  $f \in IAR(S_1 \cup \dots \cup S_i)$ . Donc,  $f \in nd(\mathcal{A}, \succeq)$ . Par conséquent,  $f \in dl\text{-Elect}(\mathcal{A}, \succeq)$ .

1.ii) Montrons à présent l'inverse. Supposons que  $f \in \mathcal{A}$  n'est pas élue et construisons une extension totale  $\succeq$  de  $\succeq$  tel que  $f \notin nd(\mathcal{A}, \succeq)$ . L'assertion  $f$  n'est pas élue signifie  $\exists C \in C(\mathcal{A}) \text{ t.q. } f \in C$  et  $\forall g \in C, f \triangleright g$  n'est pas vrai. Cela veut dire qu'il existe une extension totale  $\succeq$  de  $\succeq$  t.q.  $\forall g \in C, g \geq f$ . En effet, il suffit de définir l'ordre sur les éléments de  $C$  par rapport à  $f$  comme suit :  $\forall g \in C, g \neq f, g \geq f$ , puis de compléter le reste de la relation de sorte à étendre  $\succeq$ . Soit  $(S_1, \dots, S_n)$  la partition bien-ordonnée associée à  $\succeq$ , et soit  $f \in S_i$  pour un certain  $i \in \{1, \dots, n\}$ . Puisque  $\forall g \in C, g \geq f$ , il s'ensuit  $\forall g \in C, \text{ si } f \in S_i \text{ alors } g \in S_j \text{ pour un certain } j \leq i$ . Donc,  $\forall k \in \{1, \dots, n\}, f \notin IAR(S_1 \cup \dots \cup S_k)$ , ce qui signifie que  $f \notin nd(\mathcal{A}, \succeq)$ .

2. Pour montrer la cohérence de  $dl\text{-Elect}(\mathcal{A}, \succeq)$  par rapport à  $\mathcal{T}$ , supposons le contraire. Donc il existe un conflit  $C \subseteq dl\text{-Elect}(\mathcal{A}, \succeq)$ . Etant donné que chaque élément de  $C$  est élu, alors  $\forall f \in C, \exists g \in C, g \neq f, \text{ t.q. } f \triangleright g$ , ce qui est impossible.  $\square$

Considérons la version modifiée suivante de notre exemple.

**Exemple 7** Soient  $Reg_1, Reg_2$  et  $Reg_3$  des régions d'origine de danses.

Soit  $\mathcal{T}' = \mathcal{T} \cup \{Mdanse \sqsubseteq Reg_1, Tdanse \sqsubseteq Reg_2, Reg_1 \sqcap Reg_2 \sqcap Reg_3 \sqsubseteq \perp\}$ .

Le premier (resp. second) axiome indique que les danses modernes (resp. traditionnelles) proviennent de la région 1 (resp. région 2). Le troisième axiome énonce qu'une danse peut être originaire de deux régions mais pas de trois. Cet axiome représente un conflit ternaire, donc il ne peut pas être exprimé en DL-Lite.

Soit  $\mathcal{A}' = \mathcal{A} \cup \{Reg_1(d_6), Reg_2(d_6), Reg_3(d_6), Mdanse(d_6), Tdanse(d_6)\}$ , où  $d_6$  représente une danse.

Supposons que  $\succeq$  sur  $\mathcal{A}'$  définit les préférences strictes :

- $Reg_1(d_6) \triangleright Reg_2(d_6) \triangleright Reg_3(d_6)$ , et
- $Mdanse(d_6) \triangleright Tdanse(d_6)$ .

L'ensemble des conflits est :

$$C(\mathcal{A}') = C(\mathcal{A}) \cup \left\{ \begin{array}{l} \{Reg_1(d_6), Reg_2(d_6), Reg_3(d_6)\}, \\ \{Mdanse(d_6), Tdanse(d_6)\} \end{array} \right\}$$

Il est aisé de vérifier que :  $dl\text{-Elect}(\mathcal{A}', \succeq) = Elect(\mathcal{A}, \succeq) \cup \{Reg_1(d_6), Reg_2(d_6), Mdanse(d_6)\}$ .  $\square$

D'un point de vue calculatoire, la tractabilité peut être préservée dans le contexte de langages plus expressifs que DL-Lite à condition que le calcul des conflits se fasse de manière efficace. En effet, la complexité pour calculer la réparation  $dl\text{-Elect}(\mathcal{A}, \succeq)$  dépend de la complexité de calculer les conflits. Si cette dernière est polynomiale, alors tout le processus est polynomial. Notons que vérifier si une assertion  $f \in \mathcal{A}$  est élue revient simplement à parcourir tous les conflits assertionnels, et vérifier pour chaque conflit  $C \in C(\mathcal{A})$  s'il existe une assertion  $g \in C$  qui est strictement moins préférée que  $f$ . Ceci se fait en temps polynomial par rapport à la taille et au nombre de conflits.

## 7 Conclusion

Nous avons étudié le problème de la restauration de la cohérence d'une base ABox partiellement préordonnée et incohérente par rapport à la base TBox pour des ontologies DL-Lite. Nous avons proposé une méthode appelée Elect qui généralise la sémantique IAR (ABox non-ordonnée) et la sémantique dite non-contestée (ABox totalement préordonnée). Dans la méthode Elect, un préordre partiel est interprété comme une famille de préordres totaux auxquels l'inférence non-contestée est appliquée, produisant des réparations non-contestées. Nous avons introduit le concept d'assertions élues et proposé une caractérisation équivalente de la méthode Elect. En particulier, nous avons montré que la complexité de la méthode Elect est polynomiale.

Dans le cadre du projet AniAge, nous envisageons d'appliquer la méthode Elect à des ontologies représentant des danses de l'Asie du Sud-Est. Des experts annotent des vidéos de danses par rapport à la base TBox pour capturer les

connaissances culturelles représentées par certaines postures de danses, des mouvements, des tenues ou des accessoires. Différents experts peuvent annoter une même vidéo, donnant lieu à des conflits que nous proposons de résoudre avec la méthode Elect.

**Remerciements** : Les auteurs remercient les relecteurs pour leurs commentaires utiles à l'amélioration de cet article. Ce travail a été financé par le projet européen H2020-MSCA-RISE : AniAge (High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage).

## Références

- [1] Baader, F., D. Calvanese, D. McGuinness, D. Nardi et P. Patel-Schneider: *The Description Logic Handbook : Theory, Implementation, and Applications*. 2007.
- [2] Baget, J-F., S. Benferhat, Z. Bouraoui, M. Croitoru, M-L. Mugnier, O. Papini, S. Rocher et K. Tabia: *A General Modifier-Based Framework for Inconsistency-Tolerant Query Answering*. Dans *KR, Le Cap, Afrique du Sud*, pages 513–516, 2016.
- [3] Benferhat, S., Z. Bouraoui, H. Chadhry, M. Shafry Bin Mohd Rahim Fc, K. Tabia et A. Telli: *Characterizing Non-Defeated Repairs in Inconsistent Lightweight Ontologies*. Dans *SITIS, Naples, Italie*, pages 282–287, 2016.
- [4] Benferhat, S., Z. Bouraoui et K. Tabia: *How to Select One Preferred Assertional-Based Repair from Inconsistent and Prioritized DL-Lite Knowledge Bases?* Dans *IJCAI, Buenos Aires, Argentine*, pages 1450–1456, 2015.
- [5] Benferhat, S., D. Dubois et H. Prade: *Representing Default Rules in Possibilistic Logic*. Dans *Knowledge Representation and Reasoning*, pages 673–684, 1992.
- [6] Benferhat, S., D. Dubois et H. Prade: *Some syntactic approaches to the handling of inconsistent knowledge bases : A comparative study. Part 2 : the prioritized case*, tome 24, pages 473–511. 1998.
- [7] Benferhat, S., S. Lagrue et O. Papini: *Reasoning with partially ordered information in a possibilistic logic framework*. *Fuzzy Sets and Systems*, 144(1) :25–41, 2004.
- [8] Bienvenu, M. et C. Bourgaux: *Inconsistency-Tolerant Querying of Description Logic Knowledge Bases*. Dans *Reasoning Web : Logical Foundation of Knowledge Graph Construction and Query Answering*, tome 9885, pages 156–202. LNCS. Springer, 2016.
- [9] Bienvenu, M., C. Bourgaux et F. Goasdoué: *Querying Inconsistent Description Logic Knowledge Bases under Preferred Repair Semantics*. Dans *AAAI, Québec, Canada*, pages 996–1002, 2014.
- [10] Bienvenu, M. et R. Rosati: *Tractable Approximations of Consistent Query Answering for Robust Ontology-based Data Access*. Dans *IJCAI, Pékin, Chine*, pages 775–781, 2013.
- [11] Brewka, G.: *Preferred Subtheories : An Extended Logical Framework for Default Reasoning*. Dans *IJCAI, Detroit, USA*, pages 1043–1048, 1989.
- [12] Calvanese, D., G. De Giacomo, D. Lembo, M. Lenzerini et R. Rosati: *Tractable Reasoning and Efficient Query Answering in Description Logics : The DL-Lite Family*. *Journal of Automated Reasoning*, 39(3) :385–429, 2007.
- [13] Calvanese, D., E. Kharlamov, W. Nutt et D. Zheleznyakov: *Evolution of DL-Lite Knowledge Bases*. Dans *International Semantic Web Conference (1), Shanghai, Chine*, pages 112–128, 2010.
- [14] Cozman, F.G.: *Credal networks*. *Artificial Intelligence Journal*, 120 :199–233, 2000.
- [15] Du, J., G. Qi et Y. Shen: *Weight-based consistent query answering over inconsistent SHIQ knowledge bases*. *Knowledge and Information Systems*, 34(2) :335–371, 2013.
- [16] Dubois, D., H. Fargier et H. Prade: *Ordinal and Probabilistic Representations of Acceptance*. *Journal of Artificial Intelligence Research*, 22 :23–56, 2004.
- [17] Lembo, D., M. Lenzerini, R. Rosati, M. Ruzzi et D. Fabio Savo: *Inconsistency-Tolerant Semantics for Description Logics*. Dans *Web Reasoning and Rule Systems*, tome 6333 de LNCS, pages 103–117, 2010.
- [18] Martinez, M. V., F. Parisi, A. Pugliese, G. I. Simari et V. S. Subrahmanian: *Inconsistency Management Policies*. Dans *Knowledge Representation and Reasoning*, pages 367–377. AAAI Press, 2008.
- [19] Rescher, N. et R. Manor: *On inference from inconsistent premisses*. *Theory and Decision*, 1(2) :179–217, 1970.
- [20] Staworko, S., J. Chomicki et J. Marcinkowski: *Prioritized repairing and consistent query answering in relational databases*. *Annals of Mathematics and Artificial Intelligence*, 64(2-3) :209–246, 2012.
- [21] Telli, A., S. Benferhat, M. Bourahla, Z. Bouraoui et K. Tabia: *Polynomial Algorithms for Computing a Single Preferred Assertional-Based Repair*. *Künstliche Intelligenz*, 31(1) :15–30, 2017.
- [22] Touazi, F., C. Cayrol et D. Dubois: *Possibilistic reasoning with partially ordered beliefs*. *Journal of Applied Logic*, 13(4) :770–798, 2015.
- [23] Trivela, D., G. Stoilos et V. Vassalos: *Querying Expressive DL Ontologies under the ICAR Semantics*. Dans *DL workshop. Tempe, USA*, 2018.