Validation et révision

Validation versus revision

F. Dupin de Saint-Cyr[1] S. Loiseau[2]
LERIA
Université d'Angers, 2 bd Lavoisier, 49045 ANGERS Cedex 01
[1]bannay@info.univ-angers.fr, [2]loiseau@info.univ-angers.fr

Résumé

Cet article a pour objectif d'explorer les liens entre d'une part la validation et la mise au point des systèmes à base de connaissances et d'autre part la révision des connaissances.
La validation propose une évaluation de la qualité d'une base de connaissances ; parmi les critères qui permettent de définir cette qualité, on s'intéresse à celui de cohérence. La cohérence d'un ensemble de règles peut être vue comme la consistance potentielle de ces règles avec toutes les entrées possibles. La mise au point consiste à modifier la base de règles incohérente pour la rendre cohérente.
Un opérateur de révision est une relation d'inférence qui est définie afin de pouvoir raisonner en présence d'inconsistance, il fournit un moyen de sélectionner des règles à supprimer afin de restaurer la consistance. Dans les deux domaines, la base de connaissances initiale est généralement consistante, l'inconsistance arrive quand on ajoute à la base une entrée sur laquelle on désire raisonner. Cet article souligne les liens entre mise au point et révision en comparant la consistance à la cohérence et en définissant une relation d'inférence pour la mise au point.

Mots Clef

Validation, mise au point, cohérence, diagnostic, révision, gestion de l'inconsistance, relation d'inférence non monotone

Abstract

This paper is a first work on the link between two fields: validation and refinement in knowledge base systems and revision in knowledge representation. The validation evaluates knowledge base quality; among the criterion which can define this quality an interesting one is coherency. The coherency of a set of rules can be viewed as the potential consistency of this set of rules with respect to every possible input. Refinement is a way to modify an incoherent knowledge base in order to restore its coherency. A revision operation is an inference relation, which is defined in order to reason under inconsistency, it provides a way to select rules to delete in order to restore the consistency. In the two fields, usually the initial knowledge base is consistent and the inconsistency can only come from some input on which a deduction must be done. This paper underlines the link between refinement and revision by comparing consistency to coherency and then by defining an inference relation for refinement.

Keywords

Validation, refinement, coherency, diagnosis, revision, inconsistency handling, non monotonic inference relation

1 Introduction

De nombreux travaux ont été effectués pour fournir des critères permettant de mesurer la qualité des bases de connaissances (BC) ainsi que des algorithmes pour les évaluer. La *validation* de bases de connaissances amène à montrer le respect par une base de connaissances d'un critère donné, ce qui fournit une garantie partielle de sa qualité. Dans le cas d'une violation du critère, le développeur doit en localiser les causes afin de modifier la base de connaissances pour qu'elle le respecte. Cette modification est faite soit en supprimant ou modifiant des règles mauvaises, soit en ajoutant à la BC des contraintes sur son utilisation. La localisation des causes du non respect d'un critère et la recherche des modifications à apporter à la base constituent la *mise au point* de la base de connaissances. La validation et la mise au point ont été spécialement étudiées sur des bases de connaissances réelles, afin de fournir des outils aux développeurs. Beaucoup d'articles ont été présentés dans ce domaine [2, 26].
La validation de bases de règles divise classiquement les connaissances en deux parties : les entrées du sys-

tème et les règles. Les BC considérées dans ce papier sont des ensembles de règles qui fournissent la sémantique du domaine. Un ensemble d'entrées est une base de faits. Quand la base de règles est appliquée à un ensemble d'entrées, elle mène à des sorties (par l'utilisation d'un moteur d'inférence basé sur le modus ponens). À des fins de validation, des contraintes sont fournies : elles représentent des informations additionnelles sur le domaine qui permettent d'exclure les entrées qui ne peuvent avoir lieu ou qui associent des sorties à obtenir à partir de certaines entrées. Une contrainte particulière est un test qui fait correspondre une entrée à une sortie. Le critère le plus étudié est celui de la cohérence –une BC est cohérente si et seulement si avec n'importe quelle entrée valide, aucune contradiction ne peut être obtenue en utilisant la BC. L'étape la plus difficile à franchir est de transformer la définition informelle de la cohérence fournie ci-dessus en un critère formel et calculable. Dans cet article, nous utilisons comme critère de cohérence d'une base de règles : la VT-cohérence définie relativement à un jeu de tests donné, la VT-cohérence est acquise si on ne peut pas déduire la contradiction à partir d'une entrée "valide" ou à partir de l'entrée et de la sortie d'un test [7]. La mise au point consiste à rétablir la VT-cohérence d'une base de règles à partir d'un diagnostic. Un diagnostic contient des contraintes et des règles telles que l'ajout de ces contraintes et la suppression de ces règles restaure la VT-cohérence. Un problème difficile est celui de la sélection du meilleur diagnostic : il faut pour cela définir une relation de préférence entre diagnostics.

A coté des travaux de validation, beaucoup de recherches ont été effectuées pour gérer l'inconsistance logique d'une base de connaissances composée de formules en logiques propositionnelles. Parmi ces formules, deux types de formules peuvent être utilisés : les formules atomiques qui représentent des faits ponctuels et les formules implicatives qui représentent des règles. Contrairement à la validation des BC, les faits et les règles figurent ensemble dans la base de connaissances. Celle-ci est directement utilisée pour déduire de nouvelles informations sur le système qu'elle modélise.

La difficulté est de raisonner avec une base de connaissances inconsistante en empêchant que toute déduction ne soit triviale. Un problème particulier très étudié est celui de l'insertion d'une nouvelle information (caractérisant le système) à la base de connaissances (qui le représente). Le raisonnement après une telle insertion est appelé révision de la base de connaissances [1, 28, 15]. Ainsi, le problème de la révision est de trouver quelle formule ψ peut être déduite à partir d'une formule φ qui a été ajoutée à la base de connaissances, en d'autres mots de vérifier si φ ajoutée à la BC permet d'inférer ψ ; il faut noter qu'en toute généralité la relation d'inférence ne doit pas être classique puisque φ peut amener la BC a être inconsistante, et l'inférence classique amènerait alors à déduire n'importe quoi. C'est pourquoi beaucoup de chercheurs ont proposés des relations d'inférence non monotones pour gérer de manière raisonnable les cas d'inconsistance. Ces relations d'inférence utilisent des relations de préférence permettant de choisir des sous-théories consistantes les plus intéressantes.

Ce papier a pour objectif de comparer les deux champs de recherche que sont la validation et la révision, de façon à montrer les différences entre les théories et techniques de ces deux domaines, et de fournir des pistes de travaux pour enrichir chacun des domaines. Le papier est organisé comme suit. La première partie traite de la révision ; des exemples de relation d'inférence non monotones sont présentés (ces approches ont été définies dans [4, 12, 13]). La seconde partie présente une approche de validation et de mise au point des bases de règles. Cette approche a été présentée dans [7], la validation est basée sur un critère de cohérence issu de [20, 14], et la mise au point utilise la notion de diagnostic de [24, 16]. La troisième partie compare ces deux approches : (i) elle précise la différence entre consistance et cohérence, (ii) elle situe le processus de mise au point par rapport à celui de révision, (iii) elle fournit des perspectives de travaux que l'interaction des deux approches suggère.

2 Révision des bases de connaissances

Dans la suite du papier, nous considérons un langage propositionnel fini $\mathscr{L}$. Les éléments de $\mathscr{L}$, appelés formules, sont notés par des lettres grecques α, β,... Une interprétation pour $\mathscr{L}$ est une assignation d'une valeur de vérité de l'ensemble $\{V, F\}$ à chaque formule de $\mathscr{L}$ en accord avec les règles classiques du calcul propositionnel ; on note Ω l'ensemble de telles interprétations. Une interprétation ω est un modèle d'une formule α ($\omega \models \alpha$) ssi $\omega(\alpha) = V$. Une formule β est dite conséquence logique de α ($\alpha \models \beta$) ssi chaque modèle de α est un modèle de β. Une formule qui est satisfaite par n'importe quelle interprétation est dite une tautologie (notée $\models \alpha$). Une formule α est dite consistante ssi il en existe au moins un modèle ; une formule inconsistante est notée $\perp$. Une relation d'inférence non monotone est notée $\mid\sim$. Le problème de la révision est de décider si ayant une base de connaissances BC et une nouvelle information φ, on peut déduire ψ, ce que l'on note par $\varphi \mid\sim_{BC} \psi$.

Nous rappelons que la logique classique est dite monotone dans le sens où un résultat déduit n'est jamais remis en cause en présence de nouvelles informations. Une telle propriété n'est toutefois pas toujours souhaitable pour raisonner ; lorsque par exemple de nouvelles informations amènent à revoir des conclusions tirées précédemment. C'est pour permettre cela qu'il est nécessaire de définir des relations d'inférence non monotones autorisant un raisonnement plus général.

2.1 Définitions

De nombreuses approches ont été proposées afin de réviser une base de connaissances BC par φ. L'idée de base utilisée dans les approches que nous allons présenter est de sélectionner des sous ensembles consistants de $BC \cup \varphi$, appelés sous théories préférées, et d'utiliser l'inférence de la logique classique sur ces sous ensembles.

Définition 1 *Si $\leq$ est une relation de préférence sur des sous ensembles de formules, la relation d'inférence non monotone fondée sur $\leq$ est définie par $\varphi \mid\!\sim_{\leq,BC} \psi \Leftrightarrow \forall S$, S étant un sous-ensemble de BC consistant avec φ préféré (par rapport à $\leq$), $S \cup \varphi \models \psi$.*

L'important est de définir une relation de préférence qui permette de sélectionner les sous théories préférées les plus intéressantes. Le critère de sélection utilisé prend généralement en considération des mesures d'incertitude [27, 5] ou utilise des mesures qui prennent en compte des priorités [25, 8, 21, 4, 17] (pour une description plus détaillée de ces approches se référer à [9]). Ces approches consistent à classer les bases de connaissances en divers niveaux de priorité. La connaissance permettant de classer est en général fournie par l'expert (sauf pour le système Z [22] dans lequel le classement est fait automatiquement en privilégiant les règles les plus spécifiques). Pour préférer une sous-théorie par rapport à une autre, les approches proposées retiennent les sous-théories qui ont le plus de formules dans chacun des niveaux en commençant par le niveau de priorité le plus fort. Il faut noter que ces approches préfèrent toujours une solution contenant moins de formules mais d'un niveau de priorité élevé, à une solution contenant beaucoup de formules de faible priorité : les niveaux n'interagissent jamais.
Pinkas puis Dupin de Saint-Cyr, Lang et Schiex [23, 12, 13] proposent une approche alternative qui permet d'associer un poids (appelé pénalité) aux formules de la base de connaissances. Intuitivement, la pénalité associée à une formule représente l'importance de la formule, plus elle est grande plus la formule est importante et plus il sera difficile de la rejeter. Une formule de pénalité infinie est inviolable ; si toutes les formules sont de pénalités infinies, on se retrouve dans le cas de la logique classique. Un avantage des pénalités sur les priorités, est qu'elles sont additives dans le sens suivant : le coût associé à un sous ensemble de formules d'une base de connaissances est la somme des pénalités associées aux formules qui ont été rejetées, c'est-à-dire qui sont dans la base de connaissances mais pas dans le sous ensemble sélectionné. Le sous ensemble de plus petit coût est le sous ensemble préféré de la base de connaissances. On suppose que les pénalités sont fournies par l'expert.
Les différentes approches étudiées ont chacune leurs avantages et leur défauts ; on trouve toujours des exemples de sens communs qui sont moins bien ou mieux traités par une approche plutôt qu'une autre.
Nous allons présenter deux types de technique. Le premier ne fait pas appel à d'autres connaissances que celles contenues dans la BC mais considère des critères de maximalité en termes de quantité d'information. Le deuxième type de technique utilise des informations supplémentaires à la BC, on dispose d'information sur les importances relatives des connaissances : deux approches sont proposées soit l'utilisation d'une relation d'ordre sur les formules de la base, soit l'utilisation de pondérations de ces formules.

2.2 Exemples d'inférence à partir d'une BC classique

Les deux techniques que l'on présente consistent à sélectionner parmi les sous bases consistantes possibles $X_1, \ldots, X_n$ de la base de connaissances inconsistante, celles qui sont maximales soit du point de vue de la cardinalité soit du point de vue de l'inclusion. Dans ce dernier cas, X est une sous base maximale pour l'inclusion s'il n'existe pas une autre sous base consistante X' telle que X est incluse dans X'. Nous nous contentons d'illustrer ces techniques dans l'exemple qui suit.

Exemple 1 *([6]) Considérons une base de connaissances de laquelle on désire déduire si des personnes peuvent donner de leur temps pour du bénévolat. Les variables Salarié, PDG, Prêtre, AuFoyer sont des variables d'entrée qui caractérisent les différents types de personnes. La variable Sport représente le fait qu'une personne peut passer du temps (plus de deux heures) à faire du sport. TempsLibre est vraie si une personne a du temps libre. Bénévolat définit si la personne peut passer du temps (plus de huit heures) comme bénévole. La base de règles BC suivante décrit notre système:*

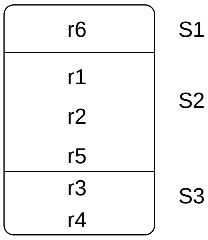
r_1 : *Salarié* $\wedge$ *PDG* $\rightarrow \neg$ *TempsLibre*
r_2 : $\neg$ *TempsLibre* $\rightarrow \neg$ *Bénévolat*
r_3 : *Prêtre* $\rightarrow$ *Bénévolat*
r_4 : *AuFoyer* $\rightarrow$ *Bénévolat*
r_5 : $\neg$ *TempsLibre* $\rightarrow \neg$ *Sport*

r_6 : *AuFoyer* $\wedge$ *Salarié* $\rightarrow \perp$

Une question est de savoir ce que l'on pourrait déduire de la base de connaissance lorsque l'on apprend les faits Salarié, PDG et Prêtre. Cela signifie que l'on recherche les formules ψ *telles que :*
Salarié $\wedge$ *PDG* $\wedge$ *Prêtre* $\mid\!\sim_{\leq,BC}\psi$*. Par définition de* $\mid\!\sim_{\leq,BC}$*, cela signifie que :* $\forall S$*,S étant une sous théorie préférée (relativement à* $\leq$*) de BC consistante avec Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre,* $S\cup\{$ *Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre*$\}$ $\models \psi$.
Si nous utilisons $\mid\!\sim_{\subseteq,BC}$ *: les sous bases préférées de BC consistantes avec la formule Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre sont :* $A = \{r_1, r_2, r_4, r_5, r_6\}$, $B = \{r_1, r_3, r_4, r_5, r_6\}$, $C = \{r_2, r_3, r_4, r_5, r_6\}$ *(elles sont aussi préférées par rapport à l'ordre fourni par la cardinalité). Chacune des 3 sous-bases préférées permet de déduire* $\neg$ *AuFoyer. Cela signifie que Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre* $\mid\!\sim_{\subseteq,BC}$ $\neg$ *AuFoyer. Toutefois on ne peut pas décider si Bénévolat est vrai ou non avec cette relation d'inférence.*

2.3 Exemples d'inférence à partir d'informations complémentaires

Si la sélection des sous bases n'est pas assez forte, des informations complémentaires fournies sur la base de connaissances peuvent permettre de trancher.
La sélection des sous-bases préférées peut prendre en compte des niveaux, soit S_1, S_2 , ... S_n les niveaux de BC rangés par priorité décroissante. Nous allons présenter trois approches utilisant ces stratifications : best-out, discrimin et lexicographique. Si A est une sous-base de BC, on appelle A_i l'ensemble des formules de A qui sont dans la strate S_i.



Définition 2 *Soit A et B deux sous ensembles consistants de BC.*

- *Soit bo(A) (pour "best-out") le degré de la strate la plus prioritaire dans laquelle une formule n'appartient pas à A :* $bo(A) = \min\{i, \exists\varphi \in S_i \setminus A\}$.

  A est dite bo-préférée à B ssi $bo(B) \leq bo(A)$ *(définition donnée dans [11])*

- *A est discrimin préférée (strictement) à B ssi* $\exists S_i$ *tel que* $B_i \subset A_i$ *et* $\forall j < i$, $A_j = B_j$ *(définition donnée dans [8])*

- *A est lexicographiquement préféré à B ssi* $\exists S_i$ *tel que* $\mid B_i \mid < \mid A_i \mid$ *et* $\forall j < i$, $\mid A_j \mid = \mid B_j \mid$ *(définition donnée dans [4])*

Exemple 2 *(suite)*

- *Dans l'exemple* $bo(A) = 3$, $bo(B) = 2$ *et* $bo(C) = 2$*. Les sous bases best-out préférées de BC consistantes avec Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre sont* $D = \{r_1, r_2, r_5, r_6\}$*, et* $A = \{r_1, r_2, r_4, r_5, r_6\}$*. Ainsi, Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre* $\mid\!\sim_{bo,BC}$ $\neg$ *AuFoyer* $\wedge$ $\neg$ *TempsLibre* $\wedge$ $\neg$ *Bénévolat* $\wedge$ $\neg$ *Sport.*

- *A est la sous-base discrimin préférée (on peut vérifier par exemple que A est bien discrimin préférée à B car* $A_1 = B_1$ *et* $B_2 \subset A_2$*) Cela signifie que : Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre* $\mid\!\sim_{dmin,BC}$ $\neg$ *AuFoyer* $\wedge$ $\neg$ *TempsLibre* $\wedge$ $\neg$ *Bénévolat* $\wedge$ $\neg$ *Sport*

- *A est également la sous-base lexicographiquement-préférée. Cela signifie que : Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre* $\mid\!\sim_{lex,BC}$ $\neg$ *AuFoyer* $\wedge$ $\neg$ *TempsLibre* $\wedge$ $\neg$ *Bénévolat* $\wedge$ $\neg$ *Sport.*

Dans l'exemple précédent, les trois relations d'inférence donnent le même résultat, ceci n'est évidemment pas souvent le cas (pour des comparaisons se reporter à [9]). On montre que :

Propriété 1 ([4]) *A bo-préféré à B* $\Rightarrow$ *A discrimin-préféré à B* $\Rightarrow$ *A lexicographiquement-préféré à B.*
La réciproque est fausse.

Au lieu d'utiliser des niveaux de priorité, les formules peuvent être pondérées par un nombre positif appelé pénalité. Pour chaque formule φ_i de la base de connaissances, sa pénalité associée α_i représente un degré de confiance en φ_i. Il peut être interprété comme le coût que l'utilisateur doit payer afin d'exclure la formule φ_i. Une pénalité infinie est associée aux formules inviolables. La relation d'inférence associée aux bases de connaissances avec pénalité est la suivante :

Définition 3 ([12]) *Le coût* $C(A)$ *d'une sous-base A de la base de connaissance BC est la somme des pénalités des formules de* $BC \setminus A$*.*
Une sous-base consistante A de BC est k-préférée à une sous base consistante B ssi $C(A) \leq C(B)$*.*

Exemple 3 *(suite) Considérons la base de connaissances à pénalité BC suivante :*

r_1 :	*Salarié* $\wedge$ *PDG* $\rightarrow$ $\neg$ *TempsLibre*	$\alpha_1 = 20$
r_2 :	$\neg$ *TempsLibre* $\rightarrow$ $\neg$ *Bénévolat*	$\alpha_2 = 20$
r_3 :	*Prêtre* $\rightarrow$ *Bénévolat*	$\alpha_3 = 5$
r_4 :	*AuFoyer* $\rightarrow$ *Bénévolat*	$\alpha_4 = 5$
r_5 :	$\neg$ *TempsLibre* $\rightarrow$ $\neg$ *Sport*	$\alpha_5 = 20$
r_6 :	*AuFoyer* $\wedge$ *Salarié* $\rightarrow \perp$	$\alpha_6 = +\infty$

Avec les sous-bases A, B, C et D des exemples 1 et 2, on obtient : $C(A) = 5$, $C(B) = 20$, $C(C) = 20$ *et* $C(D) = 10$
La sous base de BC k-préférée consistante avec Salarié $\wedge$ *PDG* $\wedge$ *Prêtre est A. Cela signifie que Salarié* $\wedge$ *PDG* $\wedge$ *Prêtre* $\mid\!\sim_{k,BC}$ $\neg$ *AuFoyer* $\wedge$ $\neg$ *TempsLibre* $\wedge$ $\neg$ *Bénévolat* $\wedge$ $\neg$ *Sport.*

3 Validation et mise au point des bases de connaissances

La validation a pour but d'estimer la qualité d'une base de connaissances de façon à suggérer si nécessaire des améliorations à l'expert. La mise au point est basée sur cette mesure de qualité, son objectif est d'aider l'utilisateur à modifier la base de connaissances pour qu'elle atteigne la qualité désirée. Dans le domaine des systèmes à bases de règles, beaucoup d'approches [20, 3, 18, 14, 6], se sont intéressées à mesurer la qualité selon un critère de cohérence.
Dans cette partie, les règles ne sont pas exactement des formules de la logique classique : chaque règle est de la forme : "R_i : Si $L_1 \wedge \cdots \wedge Ln$ alors L_{n+1}". R_i est le nom de la règle. Les L_i sont des littéraux ou "$\perp$". Parmi les règles de la base de règles BR, on distingue le sous-ensemble R des règles strictes (considérées comme sûres par l'expert et qui ne peuvent pas être modifiées), elles sont appelées *contraintes*. Les règles strictes ayant pour conséquent "$\perp$" sont appelées *contraintes d'intégrité*. Les autres règles regroupées dans le sous-ensemble r sont appelées *règles d'expertise*. Dans l'exemple ci-dessous, le nom des règles appartenant à R commence par une majuscule. Un fait est un littéral (représenté par une variable propositionnelle ou sa négation). Une base de fait, notée F, est un ensemble de faits. Les *faits en entrée* sont les faits qui peuvent être fournis par l'utilisateur pendant une session. Une entrée est une instance d'un *littéral en entrée*. L'ensemble des variables propositionnelles d'entrée est déterminé une fois pour toute par l'expert.
Nous introduisons le symbole $\vdash_{MP}$ tel que $A \vdash_{MP} \varphi$ si φ peut être déduite par modus ponens à partir de l'ensemble de formules de la logique classique associées à A (à une règle "Si $L_1 \wedge \cdots \wedge L_n$ alors L_{n+1} est associée la formule classique $L_1 \wedge \cdots \wedge L_n \rightarrow L_{n+1}$). Appliquer une base de règles BR à une base de faits F consiste à calculer l'ensemble des faits déduits par modus ponens à partir de F et de BR.
Dans le paragraphe 3.1, nous présentons la propriété de VT-cohérence [6] d'une base de règles qui fournit une mesure d'évaluation de la qualité d'une base de règles. Nous l'avons choisi parce qu'elle est représentative du critère de cohérence. Dans le paragraphe 3.2, nous présentons un moyen de restaurer la VT-cohérence d'une base, c'est-à-dire un moyen de la mettre au point.

3.1 Validation

La propriété de VT-cohérence utilise un jeu de test donné par l'expert. Formellement, un test est une paire $\langle F_{test}, S\rangle$ où F_{test} est une base de faits en entrée et S (pour sortie) est un fait. Chaque test est supposé sûr ce qui signifie que : $F_{test} \cup S \cup R \not\vdash_{MP} \perp$. Par conséquent, chaque test peut être représenté par une association logique entre une base de faits en entrée F_{test} et un fait S. Cette association logique est une contrainte associée au test.
Une base de règles est cohérente si et seulement si chaque entrée admissible ne peut pas induire de résultats contradictoires. Il y a principalement deux problèmes à résoudre : d'une part, il faut pouvoir caractériser ce qu'est une entrée admissible, d'autre part, il faut savoir déduire des résultats à partir de toute entrée admissible. Dans cet article, une entrée admissible est une base de faits en entrée qui satisfait un ensemble de contraintes. Ces contraintes sont tirées de la base de règles et du jeu de tests.

Définition 4 *Soit* $T = \langle\{f_1, \ldots, f_n\}, S\rangle$ *un test, où* $\{f_1, \ldots, f_n\}$ *est une base de faits en entrée et* S *est un fait (n'appartenant pas aux entrées), "*Si $f_1 \wedge \cdots \wedge f_n$ alors S*" est la* contrainte associée *au test, dénotée* R_T*. Soit* T_{test} *un jeu de tests, on note* R_{test} *l'ensemble des contraintes associées à tous les tests de* T_{test}*.*

Définition 5 ([18]) *Une base de faits en entrée* F_i satisfait les contraintes de R *si* $F_i \cup R \not\vdash_{MP} \perp$

Exemple 4 *Notre exemple se présente ainsi dans le formalisme décrit ci-dessus :*

r_1 : Si *Salarié* $\wedge$ *PDG* Alors $\neg$ *TempsLibre*
r_2 : Si $\neg$ *TempsLibre* Alors $\neg$ *Bénévolat*
r_3 : Si *Prêtre* Alors *Bénévolat*
r_4 : Si *AuFoyer* Alors *Bénévolat*
r_5 : Si $\neg$ *TempsLibre* Alors $\neg$ *Sport*
R_6 : Si *AuFoyer* $\wedge$ *Salarié* Alors $\perp$

Ici, la dernière règle R_6 *est une contrainte, les autres règles sont des règles d'expertise. L'ensemble des littéraux en entrée est {Salarié, PDG, Prêtre, AuFoyer, BpEnfants, BpTravail}.*
Considérons que nous disposons du jeu de tests suivant :

T_1 : $\langle$*{Salarié, PDG, BpTravail}*, $\neg$ *Bénévolat* $\rangle$
T_2 : $\langle$*{AuFoyer, BpEnfants}*, $\neg$ *Bénévolat* $\rangle$
T_3 : $\langle$*{Prêtre}*, *Bénévolat* $\rangle$
T_4 : $\langle$*{Salarié, PDG,* $\neg$ *BpTravail}*, *Sport* $\rangle$
T_5 : $\langle$*{Salarié, PDG, BpTravail}*, $\neg$ *Sport* $\rangle$

*R_{test} est composée de 5 contraintes associées à $T_1, \ldots T_5$. Par exemple, la contrainte associée à T_2 est "*Si *AuFoyer $\wedge$ BpEnfants* Alors *$\neg$ Bénévolat".*
La base F = { AuFoyer, BpEnfants, Prêtre } ne satisfait pas les contraintes $\{R_{T_2}, R_{T_3}\}$.

Intuitivement, une base de règles est cohérente si et seulement si toute entrée admissible ne permet pas de déduire de résultats contradictoires. La VT-cohérence définie ci-dessous est acquise si en associant toute entrée admissible à la base de règles BR, on ne peut pas déduire de contradiction (par modus ponens) et si en associant toute base de fait F_{test} d'un test à sa sortie S et à la base de règles BR, on ne peut pas non plus déduire la contradiction.

Définition 6 ([6]) *Une base de règles BR est* VT-cohérente *relativement à un jeu de tests T_{test} si*

1. *pour toute base d'entrée F_i satisfaisant les contraintes $R \cup R_{test}$, $\quad F_i \cup BR \not\vdash_{MP} \perp$*
2. *et pour tout test $\langle F_{test}, S\rangle \in T_{test}$, $F_{test} \cup S \cup BR \not\vdash_{MP} \perp$.*

La propriété de VT-cohérence généralise et améliore les propriétés proposées dans les travaux de [3, 18, 20].

Exemple 5 *La base de règles ci-dessus est VT-incohérente. D'une part, {Salarié, PDG, Prêtre} est une base de faits en entrée qui satisfait $R \cup R_{test}$ ({Salarié, PDG, Prêtre} $\cup R_6 \cup R_{test} \not\vdash \perp$), et d'autre part, elle permet de déclencher les règles r_1, r_2 et r_3 qui mènent à la contradiction : Bénévolat et $\neg$ Bénévolat. D'où la VT-incohérence.*
Notons qu'avec T_2 la contradiction peut être également obtenue à cause des prémisses {AuFoyer, BpEnfants} qui permettent de déclencher la règle r_4 dont la conclusion est "Bénévolat" ce qui est contradictoire avec la conclusion du test T_2 : "$\neg$ Bénévolat". C'est une autre façon de montrer la VT-incohérence. Dans certains cas seuls les jeux de tests permettent de montrer les incohérences.

3.2 Mise au point

Une fois que la base de règles est déclarée incohérente, il faut la mettre au point afin de restaurer la cohérence. Une preuve minimale de l'incohérence peut être définie comme un conflit entre une entrée et un ensemble de règles. Cependant, beaucoup de conflits peuvent coexister, c'est pour cela que la notion de diagnostic, en tant que cause possible de toutes les incohérences, a été introduite. Un diagnostic est définit par un ensemble d'entrées et de règles telles que si ces entrées et ces règles sont interdites alors le reste de la base est VT-cohérente. La modification de la base à partir des diagnostics de façon à restaurer la cohérence est appelée mise au point. Nous ne présentons pas plus précisément la notion de conflit dans cet article, pour plus de détails se référer à [6].
La VT-incohérence d'une base de règles peut être expliquée par le manque de contrainte restreignant les entrées possibles (car non admissibles pour le domaine étudié) et/ou par la présence de mauvaises règles (qui mènent à des contradictions à partir de faits admissibles). Une cause possible de la VT-incohérence est appelée diagnostic. Un diagnostic est composé de bases de faits en entrée et d'ensemble de règles d'expertise. À chaque base de faits en entrée $F_i = \{f_{i,1}, \ldots, f_{i,n}\}$ d'un diagnostic peut être associée une contrainte d'intégrité "*Si $\wedge_j$ $f_{i,j}$ Alors $\perp$*".

Remarque 1 Plus les faits sont nombreux dans la contrainte d'intégrité et moins la contrainte est restrictive, c'est pourquoi, tous les faits de F_i sont mis dans la conjonction de la contrainte d'intégrité, ainsi on perd moins d'information.

Définition 7 *Soit la paire $D = \langle E_D, r_D\rangle$, E_D étant un ensemble de bases de faits en entrée $E_D = \{\{f_{1,1}, \ldots f_{1,n}\}; \cdots; \{f_{p,1}, \ldots f_{p,m}\}\}$ et r_D étant un sous-ensemble de r (les règles d'expertise). Soit R_{ED}, l'ensemble des p contraintes d'intégrité "*Si $\wedge_j$ $f_{i,j}$ Alors $\perp$*" associées à E_D. Soit $D(BR)$ la base de règles réparée par la paire D, cette base correspond à la base BR dont les règles r_D ont été supprimées et à laquelle on a ajouté les p contraintes d'intégrité de R_{ED} : $D(BR) = (R \cup r \setminus \{r_D\}) \cup R_{ED}$.*
D est un diagnostic *de la base BR relativement à un jeu de tests T_{test} si $D(BR)$ est VT-cohérent relativement à T_{test}.*

Exemple 6 *Des diagnostics de BR sont, par exemple, $D1 = \{\varnothing, \{r_1, r_4\}\}$, $D2 = \{\varnothing, \{r_2, r_4, r_5\}\}$, $D3 = \{\varnothing, \{r_3, r_4, r_5\}\}$, $D4$ = {{ Salarié, PDG, Prêtre }, $\{r_4, r_5\}$}.*

Remarque 2 Étant donnée une base de règles BR et un jeu de tests T_{test}, il existe toujours un diagnostic, puisque R est supposé fiable et que chaque base de faits possible doit satisfaire $R \cup R_{test}$. Par exemple, la paire $\langle \varnothing, r\rangle$ est toujours un diagnostic possible.

Une base VT-incohérente devient VT-cohérente si les contraintes d'intégrité associées aux bases de faits du diagnostic sont ajoutées et si les règles du-dit diagnostic sont supprimées.

Définition 8 *Un diagnostic $\langle E_D, r_D\rangle$ est* minimal *s'il n'existe pas d'autre diagnostic $\langle E'_D, r'_D\rangle$ tel que $r'_D \subseteq$*

r_D et ($E'_D \subseteq E_D$, ou $\forall F' \in E'_D, \exists F \in E_D$ tel que $F \subseteq F'$).

Exemple 7 *- Si l'on considère les deux diagnostics $E_1 = \langle\{\{a,b\},\{a,c\}\},\{r_1\}\rangle$ et $E_2 = \langle\{\{a,b\}\}, \{r_1\}\rangle$ alors E_2 est minimal car il suffit d'interdire la conjonction des littéraux a et b et de supprimer la règle r_1 pour appliquer le diagnostic E_2 alors que deux conjonctions doivent être interdites dans E_1.*
Si l'on considère le diagnostic $E_3 = \langle\{\{a\}\},\{r_1\}\rangle$, E_1 est minimal sur l'ensemble de diagnostic $\{E_1, E_3\}$ car introduire une contrainte qui interdit a est plus restrictif qu'introduire deux contraintes qui interdisent $a \wedge b$ et $a \wedge c$ (cf. remarque 2).
- Les diagnostics donnés dans l'exemple 6 sont minimaux.

Dans [6], l'algorithme permettant de calculer les "diagnostics" d'un système physique de Reiter [24] en termes de "conflits" prouvant l'incohérence du système a été étendu pour le calcul des diagnostics minimaux d'une base de règles quand les conflits de la base sont connus. Dans notre approche, nous ne considérons que les diagnostics minimaux contenant un ensemble de règles à supprimer et un ensemble de contraintes d'intégrité à ajouter de façon à restaurer la VT-cohérence. Il peut, bien sûr, exister plusieurs diagnostics minimaux. Le choix du diagnostic minimal qui permet de réparer la base de la meilleure façon possible peut être fait en utilisant des heuristiques et/ou en faisant appel à l'expert. Ici, nous proposons de sélectionner les diagnostics minimaux qui sont non seulement consistants avec un jeu de tests mais aussi qui permettent d'obtenir la sortie attendue quand on fournit les entrées d'un test (l'idée de cette définition est exposée plus en détails dans [6]).

Définition 9 *$D1$ est un diagnostic* préféré *à $D2$ dans une base de règle BR relativement à un jeu de tests T_{test}, si $D1$ est un diagnostic minimal et si le nombre de tests validés par $D1(BR)$ est supérieur à $D2(BR)$. (Un test $\langle F_{test}, S\rangle \in T_{test}$ est validé par une base de règle B si $F_{test} \cup B \vdash_{MP} S$.)*

Cette relation de préférence entre diagnostic est basée simplement sur le nombre de tests validés, une perspective intéressante serait d'affiner ce critère en prenant en compte des informations sur l'importance relative des règles et des tests.

4 Discussion

On peut souligner deux grosses différences entre les démarches de validation-mise au point et de révision. D'une part, la validation utilise une propriété de cohérence différente de la consistance classique utilisée en révision. D'autre part, la révision fournit un formalisme permettant d'exprimer différentes façons de modifier une base de connaissances, alors que la mise au point ne propose que des solutions souvent informelles pour ces modifications.

4.1 Consistance et cohérence

À chaque règle de la base, on peut associer naturellement une implication logique. Le fonctionnement d'un moteur d'inférence pour une base de règle peut être assimilé à un démonstrateur logique basé sur le modus ponens dans la base de connaissances associée. Il semble naturel d'évaluer la cohérence d'une base de règles en testant la consistance des formules logique associées aux règles. De nombreux algorithmes ont été étudiés pour calculer la consistance d'un ensemble de formules logiques et pourraient donc être utilisés pour calculer la consistance d'un ensemble de règles. Cependant, l'existence d'un modèle pour un ensemble d'implications logiques est une condition trop faible pour évaluer la cohérence d'une base de règles.

Exemple 8 *Revenons encore à notre exemple exprimé en logique classique :*
r_1 : *Salarié $\wedge$ PDG $\rightarrow \neg$ TempsLibre*
r_2 : *$\neg$ TempsLibre $\rightarrow \neg$ Bénévolat*
r_3 : *Prêtre $\rightarrow$ Bénévolat*
r_4 : *AuFoyer $\rightarrow$ Bénévolat*
r_5 : *$\neg$ TempsLibre $\rightarrow \neg$ Sport*
r_6 : *AuFoyer $\wedge$ Salarié $\rightarrow \perp$*
Cette base de connaissances est consistante, l'interprétation {Salarié, PDG, ¬TempsLibre, ¬Bénévolat, ¬Prêtre, ¬ AuFoyer, ¬ Sport} la satisfait. Mais si on considère que {Salarié, PDG, Prêtre} est une entrée admissible, on veut pouvoir dire que BR n'est pas "cohérente", ceci est possible en utilisant la propriété de VT-cohérence.

Dans le domaine de la validation, la cohérence d'une base de règles dépend des bases de faits en entrée que l'utilisateur peut lui donner durant la phase d'utilisation de la base : ces entrées sont appelées entrées admissibles. Chaque entrée admissible peut être traduite en la conjonction de ses faits. Ce qui est intéressant dans la recherche de la cohérence de la base est de savoir si la formule logique associée à toute entrée admissible est consistante avec l'ensemble des formules associées aux règles. Des algorithmes de types ATMS [10] peuvent être utilisés à cette fin, cf. [19] par exemple.
Plusieurs difficultés apparaissent : d'une part, le coût d'une telle approche est proportionnel au nombre d'entrées admissibles ; d'autre part, le nombre de ces entrées admissibles peut être infini ; de plus, il faut trou-

ver un moyen d'obtenir ces entrées ou une modélisation de ces entrées admissibles : dès que l'on ne considère pas les jeux de tests comme les seules entrées admissibles il n'y a pas de solution efficace et simple à ce problème. Donc, les travaux sur la consistance en logique classique peuvent fournir des résultats utiles mais non suffisants pour fournir une solution complète et correcte au problème de la cohérence. Ceci explique l'introduction de propriété comme la VT-cohérence.

Notons cependant que dans la VT-cohérence, on ne teste pas exactement la consistance d'un ensemble de formules mais on vérifie que cet ensemble de formules ne permet pas d'inférer la contradiction par modus ponens. Or, si deux formules a et b sont telles que $a, b \vdash_{MP} \perp$, elles sont effectivement inconsistantes. Mais, quand $a, b \not\vdash_{MP} \perp$, ces formules peuvent quand même être inconsistantes. Cela signifie que la VT-cohérence ne permet pas de détecter toutes les inconsistances potentielles! Par ailleurs, le fait d'avoir une définition syntaxique de la VT-cohérence est très utile pour la calculabilité puisque cela permet d'utiliser un moteur d'inférence basé sur le modus ponens ce qui coûte moins cher en temps qu'une définition sémantique qui nécessiterait d'évaluer les valeurs de vérité de toutes les interprétations possibles.

4.2 La mise au point à partir des diagnostics est-elle une révision?

Comme nous l'avons dit plus haut, calculer la consistance potentielle d'une base de règles ne peut pas être envisagé raisonnablement car le nombre des entrées possibles peut être très grand voire infini. Donc, pour la détection de l'incohérence le domaine de la révision ne peut pas apporter beaucoup d'aide au domaine de la validation. Cependant, une fois que l'on sait que la base de règles est incohérente, la révision peut aider à la modifier. Déterminons si la mise au point proposée ci-dessus peut permettre de définir un opérateur de révision.

Définition 10 *Étant donnée une base de règles BR et un jeu de tests T, définissons $\mid\!\sim_{diag}$ telle que :*

$$\varphi \mid\!\sim_{diag,BR,T} \psi \Leftrightarrow$$

φ est une conjonction de faits en entrée satisfaisant les contraintes $R \cup R_{test}$, et ψ une conjonction de faits en sortie
- si BR est VT-cohérente alors $\varphi \cup BR \vdash_{MP} \psi$
- sinon pour tout diagnostic préféré $D = \langle E_D, r_D \rangle$,
$\left\{ et \begin{array}{l} \varphi \text{ satisfait les contraintes } R \cup R_{test} \cup R_{ED} \\ \varphi \cup D(BR) \vdash_{MP} \psi \end{array} \right.$

En d'autres termes, si BR est VT-cohérente avec T alors φ permet de déduire ψ si et seulement si φ est une entrée admissible et si en lui associant BR on peut inférer classiquement ψ. Dans l'autre cas, c'est-à-dire si BR est VT-incohérente, pour tout diagnostic minimal $\langle E_D, r_D \rangle$, φ doit être admissible aussi relativement à R_{ED} et associée à BR dont on a supprimé les règles r_D, φ doit permettre d'inférer ψ par modus ponens.

Remarque 3 Ici, la relation d'inférence est définie sur des formules particulières, les formules φ et ψ doivent être données sous la forme d'une conjonction de faits : il est par exemple impossible de déduire une disjonction ou une implication, ou encore de déduire quelque chose à partir d'une implication. De plus, l'inférence ne peut prendre en compte que des entrées admissibles.

Remarque 4 La relation d'inférence prend en compte la base entière étant donnée toute entrée admissible (un test de VT-cohérence doit être effectué avant de pouvoir inférer) au lieu de considérer simplement l'impact d'une entrée donnée sur la base comme dans le cas de la révision classique. Ceci implique que le calcul des diagnostics est effectué une fois pour toutes ; après ce calcul, toute entrée peut être traitée de façon triviale. Ce n'est pas le cas en révision où pour chaque entrée la base de connaissances est modifiée afin que l'entrée soit prise en compte pour restituer la consistance de la base. Ainsi, le calcul de $\mid\!\sim_{diag}$ est plus complexe (à cause du test de VT-cohérence) que le calcul à effectuer pour une relation d'inférence non monotone comme celles présentées au paragraphe 2.2 mais le test de VT-cohérence peut être fait une fois pour toutes permettant ensuite des déductions immédiates.

Propriété 2 *La relation $\mid\!\sim_{diag}$ vérifie : si $a \mid\!\sim_{diag} b$ alors $\forall c$, soit $a \wedge c \mid\!\sim_{diag} b$ soit $\forall x$, $a \wedge c \not\mid\!\sim_{diag} x$. C'est-à-dire qu'une nouvelle information permet soit de déduire au moins les mêmes choses qu'avant soit inhibe toute nouvelle déduction.*

Démonstration : Si $a \mid\!\sim_{diag} b$ alors
- si BR est VT-cohérente alors $a \cup BR \vdash_{MP} b$ donc $(a \wedge c) \cup BR \vdash_{MP} b$
- sinon - soit pour tout diagnostic préféré D, $a \wedge c$ satisfait les contraintes $R \cup R_{test} \cup R_{ED}$, dans ce cas, puisque $a \cup D(BR) \vdash_{MP} b$ on a bien $(a \wedge c) \cup D(BR) \vdash_{MP} b$
- soit il existe un diagnostic préféré D_0, pour lequel $a \wedge c$ ne satisfait pas les contraintes $R \cup R_{test} \cup R_{ED_0}$, dans ce cas, aucune formule ψ n'est déductible de $a \wedge c$. □

Cette propriété signifie que la relation définie sur la VT-cohérence n'est pas du tout aventureuse, et que, dans le cas où elle permet de faire des déductions, elle est monotone.

Comme nous l'avons souligné dans la remarque ci-dessus, le calcul des diagnostics peut être effectué une fois pour toutes et ensuite la relation d'inférence n'est autre qu'une inférence par modus ponens basée sur une nouvelle base de règles (cette inférence étant déclenchée après vérification que les entrées satisfont les contraintes). Le calcul de la nouvelle base de règles ne dépend pas de l'entrée.

La définition d'un opérateur de révision dans le cas de la mise au point semble artificielle puisque le jeu de tests sur lequel est basé l'opérateur est donné une seule fois avant le calcul et que cet opérateur ne dépend pratiquement pas de l'entrée (on teste simplement si l'entrée est valide alors l'inférence par modus ponens est exécutée sinon rien ne peut être déduit).

L'objectif des deux domaines de la mise au point et de la révision est très différent : dans le premier cas, on désire réparer la base de connaissances une fois pour toutes, alors que dans le deuxième cas on désire conserver la base telle quelle mais pouvoir raisonner en présence de toute nouvelle information. C'est pourquoi les propriétés des relations d'inférence en validation ou en révision sont différentes.

4.3 Perspectives

Bien que la révision ne puissent pas apporter d'aide à la validation puisqu'il n'y a pas d'algorithmes permettant de calculer des inconsistances potentielles, ou incohérences, elle peut quand même être liée à la mise au point. Nous venons de voir qu'une relation d'inférence peut être définie à partir d'une base de règles BR et d'un jeu de tests T à partir du calcul des diagnostics. Un diagnostic est un ensemble de règles à supprimer associé à un ensemble de conjonctions de faits à interdire. La relation d'inférence que nous proposons est l'inférence par modus ponens à partir d'une nouvelle base de règles BR' calculée une fois pour toutes à partir de BR et de T (si BR est VT-cohérente avec T alors $BR = BR'$ sinon BR' est égale à BR dont on a supprimé des règles et à laquelle on a ajouté des contraintes d'intégrité en suivant un diagnostic donné). En fait, dans le cas d'une VT-incohérence, il y a plusieurs choix de diagnostic possibles afin de réparer BR en BR'. Dans ce papier, une relation de préférence entre diagnostics et proposée, la relation d'inférence doit examiner toutes les bases BR' correspondantes aux diagnostics préférés.

Un des aspects fondamentaux des méthodes de révision se base sur la définition de relations de préférence entre sous-théories : la richesse des recherches sur ces relations de préférences offre une perspective intéressante pour la mise au point. Ces relations de préférences pourraient raffiner la sélection des diagnostics préférés dans le domaine de la validation. On peut, par exemple, utiliser la théorie des pénalités pour classer les diagnostics. Pour cela, on peut demander à l'expert ou induire à partir de la forme des règles, une pénalité associée à chaque règle.

Un autre point intéressant est que la communauté validation travaille avec des applications réelles. L'utilisation de ces applications dans le cadre des travaux de révision permettrait une validation pratique des méthodes de révision.

Dans cet article, nous avons fait le lien entre la révision et la mise au point, nous ne pouvons terminer sans aborder une autre opération de modification d'une base de connaissances : *la mise à jour*. Winslett [28] suivie par Katsuno et Mendelzon [15] ont montré que deux opérations peuvent être utilisées pour modifier une base de connaissances quand une information y est ajoutée : soit une *révision* si l'information décrit le système lui-même (c'est-à-dire que l'arrivée de cette information traduit une évolution des connaissances sur le système), soit une *mise à jour* si l'information décrit une évolution du système. La mise à jour ne semble pas être liée à la validation ni à la mise au point puisque le système étudié est statique. Cependant, peut-être pourrait-on établir un lien entre la mise à jour et la gestion des versions d'une base de règles.

Références

[1] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change : partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] Ayel. Eurovav. In *(European Symposium on the Validation and Verification of Knowledge Based Systems)*, Chambéry, 1995.

[3] A. Beauvieux and P. Dague. A general consistency (checking and restoring) engine for knowledge bases. In *European Conference on Artificial Intelligence*, pages 77–82, 1990.

[4] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax- based entailment. In Ruzena Bajcsy, editor, *Proc. of the 13^{th} IJCAI*, pages 640–645, Chambery, France, 1993. Morgan-Kaufmann.

[5] S. Benferhat and P.Smets. Belief functions for logical problems : representing default rules in ε-beliefs logics. In *Abstract of the Dagstuhl Seminar*, Dagstuhl, Germany, September 1993.

[6] Bouali, Loiseau, and Rousset. Revision of rule bases. In *EUROVAV*, 1997.

[7] F. Bouali. *Diagnostic, Validation et Réparation de bases de connaissances.* PhD thesis, Paris-XI, 1996.

[8] G. Brewka. Preferred subtheories : An extended logical framework for default reasoning. In N.S. Sridharan, editor, *Proc. of the 11^{th} IJCAI*, pages 1043–1048, Detroit, Michigan, 1989. Morgan-Kaufmann.

[9] C. Cayrol and M.C. Lagasquie-Schiex. Nonmonotonic syntax-based entailment: a classification of consequence relations. In C. Froidevaux and J. Kohlas, editors, *Lecture Notes in Artificial Intelligence 946 (Proc. of ECSQARU-95)*, pages 107–114, Fribourg, Switzerland, 1995. Springer Verlag.

[10] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[11] D. Dubois, J. Lang, and H. Prade. Inconsistency in possibilistic knowledge bases - to live or not live with it. *Fuzzy logic for the management of uncertainty*, pages 335–351, 1992.

[12] F. Dupin de Saint-Cyr, J. Lang, and T. Schiex. Gestion de l'inconsistance dans les bases de connaissances : une approche syntaxique basée sur la logique des pénalités. In *Proc. du $9^{ième}$ congrès Reconnaissance des Formes et Intelligence Artificielle*, Paris, France, January 1994.

[13] F. Dupin de Saint-Cyr, J. Lang, and T. Schiex. Penalty logic and its link with Dempster-Shafer theory. In *Proc. of the 10^{th} Conf. on Uncertainty in Artificial Intelligence*, pages 204–211. Morgan Kaufmann, July 1994.

[14] Ginsberg and Williamson. Inconsistency and redundancy checking for quasi-first-order-logic knowledge bases. *International Journal of Expert Systems*, 6(3):321–340, 1993.

[15] H. Katsuno and A.O. Mendelzon. On the difference between updating a knowledge base and revising it. In J. Allen and al., editors, *Proc. of the 2^{nd} Inter. Conf. on Principles of Knowledge Representation and Reasoning*, pages 387–394, Cambridge, MA, 1991.

[16] Console L., Friedrich G., and Dupré D.T. Model-based diagnosis meets error diagnosis in logic programs. In *International Joint Conference on Artificial Intelligence*, pages 1494–1499, Chambery, France, 1993.

[17] D. Lehmann. Another perspective on default reasoning. Technical Report 92-12, Leibniz Center for Research in Computer Science. Hebrew University of Jerusalem, Israel, July 1992.

[18] Loiseau. Refinement of knowledge bases based on consistency. In *European Conference on Artificial Intelligence*, pages 845–849, 1992.

[19] Loiseau. Validation, mise au point et interaction, rapport d'h.d.r. Technical Report 1150, L.R.I., Université de Paris Sud, Orsay, 1998.

[20] Rousset M.C. On the consistency of knowledge bases: the covadis system. In *European Conference on Artificial Intelligence*, pages 79–84, 1988.

[21] B. Nebel. Belief revision and default reasoning: Syntax-based approaches. In J.A. Allen, R. Fikes, and E. Sandewall, editors, *Proc. of the 2^{nd} KR*, pages 417–428, Cambridge, MA, 1991. Morgan-Kaufmann.

[22] J. Pearl. System Z : a natural ordering of defaults with tractable applications for default reasoning. In M. Vardi, editor, *Proceedings of theoretical aspects of reasoning about knowledge*, pages 121–135, San Mateo, 1990. Morgan Kaufman.

[23] G. Pinkas. Propositional nonmonotonic reasoning and inconsistency in symmetric neural networks. In *Proc. of the 12^{th} IJCAI*, pages 525–530, Sydney, Australia, 1991. Morgan-Kaufmann.

[24] Reiter R. A theory of diagnosis from first priciples. *Artificial Intelligence*, 32:57–95, 1987.

[25] N. Rescher. *Hypothetical reasoning*. North-Holland, 1964.

[26] VanHarmelen and VanVelde. Eurovav. In *(European Symposium on the Validation and Verification of Knowledge Based-Systems)*, Leuven, 1995.

[27] N. Wilson. Default logic and Dempster-Shafer theory. In *Proc. of the 2^{nd} European Conference on Symbolic and Qualitative Approaches to Reasoning and Uncertainty*, pages 372–379, Granada, Spain, 1993. Springer-Verlag.

[28] M. Winslett. Reasoning about action using a possible models approach. In *Proc. of the 7^{th} National Conference on Artificial Intelligence*, pages 89–93, St. Paul, 1988.