ANTOINE NOËL DE TILLY

Le raisonnement à base de logique propositionnelle à l'appui de la fusion et de la révision de bases de données géospatiales

Mémoire présenté
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de maîtrise en sciences géomatiques
pour l'obtention du grade de Maître ès sciences, (M.Sc.)

FACULTÉ DE FORESTERIE ET DE GÉOMATIQUE
UNIVERSITÉ LAVAL
QUÉBEC

décembre 2007

Résumé

Le but de ce mémoire était d'effectuer, dans un contexte géospatial, une comparaison d'une approche de raisonnement qualitatif basée sur le *PROLOG* avec une autre approche reposant sur l'*ASP*. La principale question que nous posons est la suivante : Le moteur de raisonnement *Smodels* rendant possible la mise en oeuvre du raisonnement non monotone poussé et faisant intervenir le concept de modèle stable peut-il nous permettre de résoudre des problèmes de vérification de cohérence ontologique et des problèmes de révision dans le contexte de la géomatique ? Pour y répondre, nous avons procédé à une série de tests sur un échantillon de la Base nationale de données topographiques (BNDT). À la lumière des résultats obtenus, cette approche se montre très efficace et contribue à l'amélioration de la cohérence de l'information géospatiale et du raisonnement spatial réalisé à partir de cette dernière.

Abstract

The objective of this thesis is to make a comparison between a qualitative reasoning approach based on *PROLOG* with another approach based on ASP. Our principal research question was the following : Can the *Smodels* reasoning engine, allowing for advanced non monotonic reasoning and introducing the stable model concept, allow us to solve ontological consistency checking problems as well as revision problems in a geomatic context ? To answer this question, we carried out a series of tests on a cross-section from the National Topographical Database (NTDB). In the light of the results obtained, this approach has proven very effective and contributes to the amelioration of geospatial information consistency and to the resultant improvement in spatial reasoning.

Avant-propos

Ce mémoire est le fruit de nombreux mois de travail assidu qui ont permis d'établir un lien entre la logique et la géomatique. Il n'aurait pu être accompli sans la collaboration et le soutien de nombreuses personnes qui m'ont épaulé avant et pendant sa rédaction.

Ainsi, j'aimerais exprimer ma grande reconnaissance à l'égard de mon directeur Geoffrey Edwards qui m'a fourni l'occasion de participer à des activités de recherche en géomatique dès le début de mon baccalauréat et qui m'a encouragé à réaliser une maîtrise. Un grand merci pour les conseils, le soutien, et les nombreuses discussions !

Je souhaite également remercier grandement mon codirecteur Mir Abolfazl Mostafavi pour les nombreuses rencontres, ses conseils et son soutien.

Je remercie tout particulièrement Madame Odile Papini de m'avoir donné une formation sur la logique lors d'un séjour à Québec, de m'avoir encadré lors d'un stage à l'Université du Sud - Toulon - Var et d'avoir pris le temps de répondre à plusieurs de mes questions sur la logique et d'avoir annoté minutieusement mon premier manuscrit.

Je remercie Monsieur Jean Brodeur d'avoir accepté de réaliser l'évaluation de mon mémoire.

Je tiens également à remercier ma compagne Mireille qui m'a soutenu au cours de la rédaction de ce mémoire.

Aussi, je remercie ma mère Zita, mon père Robert et ma soeur Ariane de m'avoir encouragé à persévérer pendant la rédaction de ce mémoire.

Finalement, je tiens également à offrir ma reconnaissance à toutes les personnes qui m'ont aidé dans mon cheminement et dont je n'ai pas mentionné le nom.

Glossaire

0.1 Logique propositionnelle

0.1.1 Notions syntaxiques de la logique propositionnelle

Littéral : Un littéral constitue une proposition ou la négation d'une proposition.

Proposition : Énoncé déclaratif auquel il est possible d'associer une valeur de vérité, soit vraie, soit fausse.

Variable propositionnelle : Variable constituant une proposition.

Langage propositionnel fini ($\mathcal{L}$) : Langage formé d'un ensemble dénombrable de propositions et lequel ensemble est fini dans le contexte de l'informatique.

$\neg$: Connecteur de négation

$\vee$: Connecteur de disjonction

$\wedge$: Connecteur de conjonction

$\perp$: Contradiction

$\top$: Tautologie

$\rightarrow$: Implication

$\leftrightarrow$: Équivalence

Clause : Expression logique formée d'une disjonction de littéraux qui constituent des propositions.

Formule : Expression logique qui est soit une variable propositionnelle, soit une constante tautologique (0 ou 1) ou soit une expression formée de connecteurs logiques et de variables propositionelles telles que $P \vee Q$, $\neg P \wedge Q$, $P \rightarrow Q$, $\neg P \leftrightarrow Q$.

Atome : Formule qui ne contient pas de sous-formules.

Théorie : Ensemble de formules déductivement clos.

Axiome : Vérité qui est admise sans démonstration.

Théorème : Formule pour laquelle il existe une déduction à partir d'axiomes.

0.1.2 Notions sémantiques de la logique propositionnelle

Interprétation : Fonction w associant à chaque variable propositionnelle d'un langage propositionnel fini une valeur de vérité, soit 1 ou 0.

De manière formelle, nous appelons interprétation, toute application σ de P dans $\{0, 1\}$ telle que $\sigma(0)$=0 et $\sigma(1)$=1. L'application σ s'étend aux formules tel qu'il suit :

- $\sigma(\neg P)$ = 1 - $\sigma(P)$;
- $\sigma(P \vee Q) = max(\sigma(P), \sigma(Q))$;
- $\sigma(P \wedge Q) = min(\sigma(P), \sigma(Q))$;
- $\sigma(P \rightarrow Q) = max((1 - \sigma(P)), \sigma(Q))$;
- $\sigma(P \leftrightarrow Q) = min(max((1 - \sigma(P)), \sigma(Q)), max(\sigma(P), (1 - \sigma(Q))))$.

0.2 Logique des prédicats

0.2.1 Notions syntaxiques de la logique des prédicats

Symbole fonctionnel : Symbole représentant une fonction constituant un terme.

Terme :

- Une variable x est un terme ;
- Un symbole fonctionnel f est un terme ;
- Si $t_1, ..., t_n$ sont des termes, $f(t_1, ..., t_n)$ est un terme.

Atome :

- Si $t_1, ..., t_n$ sont des termes et P est un prédicat alors $P(t_1, ..., t_n)$ est un atome.

Vocabulaire : Ensemble constitué de symboles prédicatifs aussi appelés prédicats, de symboles fonctionnels, de variables, de connecteurs et de quantificateurs.

$\forall$: Quantificateur universel.

$\exists$: Quantificateur existentiel.

$\neg$: Connecteur de négation

$\vee$: Connecteur de disjonction

$\wedge$: Connecteur de conjonction

$\perp$: Contradiction

$\top$: Tautologie : Expression logique toujours vraie.

$\rightarrow$: Implication

$\leftrightarrow$: Équivalence

Clause : Disjonction de littéraux qui sont des atomes. Ex : $A(t_1) \vee \neg A(t_1) \vee \neg B(t_2)$.

Formule : Énoncé logique composé d'un ou plusieurs atomes. Ex : *A(t1)* .

- un atome est une formule ;
- si A et B sont des formules alors $\neg A$, $A \vee B$, $A \wedge B$, $A \leftrightarrow B$ sont des formules ;
- Si A est une formule et x une variable alors $\forall x A$, $\exists x A$ sont des formules.

Formule bien formée : Formule ou expression logique qui respecte les contraintes imposées par la syntaxe du langage.

Variable libre : Si l'occurence d'une variable est placée sous la portée d'un quantificateur, cette occurence est considérée liée, autrement elle est considérée libre.

Formule close ou fermée : Formule qui ne comporte pas de variable libre.

0.2.2 Notions sémantiques de la logique des prédicats

Modèle : Interprétation qui rend une formule vraie.

Théorie : Ensemble de formules.

Mod(φ*)* : Ensemble des interprétations qui satisfont une formule φ.

$w \models \varphi$: w est un modèle de φ .

φ, ψ, μ,... sont des formules de $\mathcal{L}$.

Interprétation : Une interprétation constitue une fonction associant à chaque variable propositionnelle de $\mathcal{L}$ la valeur 1 ou la valeur 0 comme suit :

- Si x est une variable libre alors $I(x) = I_v(x)$;
- $I(f(t_1, ..., t_n))$= $(I_c(f))(I(t_1), ..., I(t_n)$;
- $I(P(t_1, ..., t_n))$= $(I_c(P))(I(t_1), ..., I(t_n)$;
- Si A et B sont des formules alors $\neg A$, $A \vee B$, $A \wedge B$, $A \leftrightarrow B$ doivent être interprétés comme des formules de logique propositionnelle ;
- Si A est une formule et x une variable alors $I(\forall x) = 1$ si $I_{x/d}(A) = 1$ pour tout élément $d \in D$;
- Si A est une formule et x une variable alors $I(\exists x) = 1$ si $I_{x/d}(A) = 1$ pour tout élément $d \in D$.

Mod(φ*)* : Ensemble des interprétations qui satisfont une formule φ.

Σ : Ensemble de formules (ou base de croyances).

Cn(Σ*)* : Ensemble des conclusions pouvant être obtenues à partir d'un ensemble de formules Σ (Fermeture déductive).

0.2.3 Aspect axiomatique de la logique des prédicats

Axiomes de l'extension à la logique des prédicats du système de Lukasiewisz :

$(P \rightarrow (Q \rightarrow P))$;

$(P \rightarrow (Q \rightarrow R)) \rightarrow (P \rightarrow Q) \rightarrow (P \rightarrow R)$;

$((\neg P \rightarrow \neg Q) \rightarrow (Q \rightarrow P))$;

$(\forall x P(x) \rightarrow P(t))$;

$((S \rightarrow Q) \rightarrow (S \rightarrow \forall x Q))$.

Le modus ponens se définit de même la manière qu'en logique propositionnelle, soit tel qu'il suit : $\frac{\vdash P,\, \vdash P \rightarrow Q}{\vdash Q}$

La règle de généralisation s'exprime de la façon suivante :

$\frac{\vdash P}{\vdash \forall x P}$

0.3 La logique possibiliste

π : Distribution de possibilité.

$\pi(\omega)$: Degré de possibilité lié à chaque interprétation w.

Ω : Ensemble d'interprétations.

Π_π : Degré de possibilité associé à la distribution de possibilité π.

N_π : Degré de nécessité. Le degré de nécessité évalue à quel point φ peut être inféré par π.

$\models_\pi$: Inférence sémantique issue de π.

K_{pos} : Base de croyances possibiliste.

$\alpha - coupe$: Ensemble des formules pondérées de K_{pos} dont le poids est supérieur à α. Il se note : $K_{pos,\geq\alpha}$.

$Inc(K_{pos})$: Degré d'incohérence associé à K_{pos}.

Inférence sémantique possibiliste : Soit π une distribution de possibilité normalisée et φ une formule et $min\{\pi(\omega) : \omega \in \Omega\} = \{\omega \in \Omega : \nexists \omega' \in \Omega : \pi(\omega) < \pi(\omega)'\}$. La formule φ peut être inférée sémantiquement de π, ce qui se note $\models_\pi \varphi$, si et seulement si :

$\models_\pi$ si et seulement si $\forall \omega, \omega'$, $min\{\pi(\omega) : \omega \in \Omega\}$ nous avons $\omega \in Mod(\varphi)$.

0.4 Terminologie de *PROLOG* et d'Answer Set Programming*ASP*

Fait : Expression logique traduisant une réalité de nature factuelle.

Règle : Expression logique permettant de déduire des informations à partir d'ensembles de faits.

Requête : Opération d'interrogation effectuée à l'intérieur d'un programme de *PROLOG* permettant de vérifier la validité d'un fait à l'intérieur d'un programme.

not : Opérateur de négation par échec. Par exemple, si nous avons l'expression *not p*, cela signifie que nous ignorons si p est vrai et que, jusqu'à preuve du contraire, p est supposé faux.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

Introduction

1.1 Mise en situation

Depuis des siècles, l'humain est amené à raisonner sur son environnement. Dans cet objectif, il est amené à tirer des interprétations qualitatives sur l'espace physique afin de se situer par rapport à lui. Il est régulièrement conduit à caractériser les objets géographiques qui l'entourent de même qu'à définir des relations spatiales entre ces objets. Dans de nombreuses situations, il tâche de situer les objets autour de lui, les uns en lien avec les autres. Fréquemment, il raisonne qualitativement sur des informations à caractère géographique et tente de comprendre et de définir des relations entre de tels objets en se servant d'outils telles que les cartes afin d'interpréter l'espace physique.

L'intérêt pour les relations spatiales en géomatique s'accorde bien avec les interrogations auxquelles les humains sont confrontés dans leur vie de tous les jours. Fréquemment, l'humain est amené à se poser des questions du type : Existe-t-il une jonction entre une route et une certaine autoroute ? Quel chemin routier devrais-je emprunter pour me rendre de Québec à Vancouver ? Existe-t-il un chemin qui minimise la distance pour se rendre d'une destination à une autre ? Existe-t-il un chemin qui permet de maximiser l'utilisation de l'autoroute pour se rendre à un endroit précis ? Ainsi, l'humain se retrouve dans des panoplies de situations de la vie courante où il doit raisonner sur l'espace physique. Pour parvenir à effectuer de tels raisonnements dans des situations complexes, il se tourne vers l'utilisation de modèles physiques afin de pouvoir tirer des interprétations sur son environnement. D'un point de vue matériel, le développement des ordinateurs a permis de rendre traitables des problèmes qui ne l'étaient pas à l'origine, étant donné leur degré de complexité, et d'améliorer l'efficacité calculatoire lors de la réalisation d'opérations d'analyses spatiales et également de traiter des volumes de données de plus en plus grands.

La géomatique est un domaine de recherche en constante évolution qui s'intéresse à la cueillette, à l'entreposage et à l'interprétation de l'information spatiale propre à l'espace physique. Les applications de la géomatique sont aujourd'hui aussi nombreuses que diversifiées et leurs besoins se font de plus en plus manifestes au sein des entreprises. Les besoins de nombreuses entreprises dans des domaines tels que la géomatique et les technologies de l'information tendent à croître et à se diversifier, et ce, de manière de plus en plus prononcée. Afin de répondre à leurs besoins, ces entreprises sont amenées à collecter et à traiter des données multisources qui servent à tirer des interprétations sur le monde physique mais dont le volume tend à augmenter constamment, ce qui engendre des problèmes à la fois de stockage et de traitabilité de l'information spatiale.

Les difficultés engendrées par la gestion de l'information spatiale se situent dans un premier temps au niveau conceptuel, ce qui a donné naissance à de nombreux travaux sur les ontologies dans différents domaines scientifiques. Dans un deuxième temps, elles apparaissent lorsque des grands nombres de relations sont définis entre des objets de différentes natures suivant le domaine d'application. En géomatique, les informations spatiales portant sur des objets spatiaux, en plus d'être souvent volumineuses, ont la propriété de généralement comporter une grande quantité d'exceptions. Également, il existe habituellement de nombreuses interdépendances entre ces informations. À titre d'exemple, ces objets spatiaux peuvent être des routes, des autoroutes, des cours d'eaux, des bâtiments et autres. De plus, les relations spatiales qui lient de tels objets sont susceptibles de comporter des incohérences. Ainsi, certaines relations de dépendance interne peuvent conduire à des ensembles de relations spatiales incohérentes entre elles. Afin de surmonter ces difficultés, plusieurs chercheurs se sont penchés sur elles, les problèmes conséquemment engendrés, et ont proposé de nouveaux concepts de modélisations, ainsi que de nouvelles approches de résolution.

Une façon de raisonner qualitativement sur des informations [1] spatiales et ainsi de déceler celles qui sont incohérentes entre elles est de recourir à des représentations logiques de l'information spatiale pour traiter celles-ci à un niveau ontologique. Le fait de raisonner qualitativement sur des informations spatiales à un niveau ontologique et de réviser de telles informations de manière à les rendre cohérentes entre elles dans un premier temps comporte l'avantage d'éviter certaines erreurs de calcul lors d'analyses spatiales effectuées ultérieurement au niveau des données impliquant du raisonnement quantitatif. Ainsi, raisonner qualitativement dans un premier temps sur des ontologies peut permettre d'éviter de raisonner avec de fausses valeurs lorsque vient dans un

[1] Dans le cadre de ce mémoire, nous utilisons régulièrement la forme plurielle du mot «information» afin d'insister sur la multiplicité des informations et sur la diversité de leurs sources.

deuxième temps le moment de raisonner quantitativement sur des informations spatiales au niveau des données. Toutefois, avant d'en arriver à raisonner sur des informations à un niveau ontologique, il importe de bien conceptualiser et représenter les connaissances propres au domaine étudié.

Dans le but de relever ce défi, plusieurs chercheurs ont réalisé des travaux sur le sujet dans les domaines de la géomatique, de la topologie et de l'intelligence artificielle [EH90], [RCC92],[MEJ04b], [BE01], [SM01], [MEJ04a], [Gru93b], [Gru93a]. Étant donné la diversité des objets géographiques et la quantité croissante d'information spatiale, des chercheurs se sont attardés à des questions de catégorisation de définition de concepts propres à ces objets. Ceci a conduit à la réalisation de plusieurs travaux sur les ontologies dans le domaine spécifique de la géomatique [SM01], [MEJ04b], [BE01].

Les progrès de la géomatique dans les 30 dernières années ont beaucoup contribué au raffinement de ces interprétations. Également, au fil de ces années, les théories sur les relations spatiales et la topologie en géomatique se sont développées considérablement, entre autres, avec l'avènement des travaux de Max J. Egenhofer [EH90] et ceux de A. G. Cohn [RCC92]. Ce développement a permis de faciliter la réalisation d'interprétations dans notre environnement physique.

L'intelligence artificielle constitue aujourd'hui un domaine en croissance soutenant de plus en plus le développement et l'application d'outils d'analyse et de raisonnement en géomatique. Celle-ci permet d'automatiser des processus de raisonnement spatial, ce qui en fait un outil de grande utilité. Une des difficultés importantes liée au raisonnement spatial est de parvenir à réaliser des raisonnements qui sont exacts étant donné la présence probable d'incohérences au sein des informations géographiques que possède son utilisateur. Or, des opérations de vérification de cohérence, de fusion et de révision de croyances qui permet de rétablir la cohérence au sein d'un ensemble d'informations, peuvent permettre de parer à ce problème. Plusieurs travaux ont été menés sur ces opérations, dont [MEJ04b], [BLP05], [WPJ00], [Wur00] et [Lag03].

L'intelligence artificielle est un support intéressant pour recréer des processus de raisonnement qualitatif à l'intérieur d'implantations. Ainsi, dans ce domaine, Cohn a développé le RCC8 [Coh96], [RCC92] un calcul relationnel permettant de représenter l'information spatiale et topologique.

Parallèlement, sensiblement dans la même période, Egenhofer a développé le modèle des 9 intersections [EH90]. Ce modèle permet de caractériser les différentes manières suivant lesquelles des objets peuvent entrer en relations topologiques entre eux. Son modèle permet de caractériser les relations topologiques pouvant exister dans des objets géographiques et d'avoir une bonne compréhension des relations géométriques qui existent entre de tels objets.

De manière particulière, ce sont les travaux récents de Mostafavi et Edwards [MEJ04b] qui retiennennent plus spécialement notre attention. Au cours des denières années, ces chercheurs ont développé une méthode de vérification de cohérence ontologique pour l'information spatiale s'appuyant sur une approche de raisonnement qualitatif. Cette méthode a nécessité la création d'une approche poussée pour la représentation d'ontologies en *PROLOG*, un langage de programmation logique. Le *PROLOG* s'est montré très efficace pour la vérification de cohérence ontologique, hormis dans quelques exceptions qui ont posé des limites auxquelles nous porterons attention. De telles limites peuvent éventuellement être rencontrées dans des bases de données utilisées par des entreprises et des organismes gouvernementaux.

Depuis plusieurs années, les bases de données constituent des outils de gestion très précieux, et ce, particulièrement avec les données spatiales. De nombreuses entreprises et organismes gouvernementaux oeuvrant dans une panoplie de domaines recourent à ce genre d'outil pour suivre l'évolution de leurs activités et améliorer la gestion de leurs ressources. Ceci s'effectue, entre autres, en veillant au maintien de la cohérence d'une base de données. Si une base de données qui n'évolue pas, en termes de contenu, peut facilement conserver sa cohérence interne, un tel type de base de données demeure très rare dans notre monde. En fait, les bases de données subissent habituellement des ajouts et des suppressions d'informations sans que celles-ci ne soient actualisées. Toutefois, lors de l'ajout d'informations supplémentaires au cours de l'évolution du temps, des processus de vérification de cohérence et de révision des croyances fréquents s'imposent afin de garantir le maintien de cette cohérence. Les croyances sont des représentations évolutives de faits représentant la réalité. Aussi, une base de données traditionnelle ne suffit plus en tant que support représentationnel pour permettre de réussir à bien accomplir un processus de révision d'informations géographiques. La révision est une opération qui vise à permettre la préservation ou la restauration de la cohérence de ces dernières. Certaines approches de raisonnement logique présentent un intérêt dans un tel contexte. Il devient alors intéressant de recourir à des «bases de géocroyances» qui sont essentiellement le résultat de la traduction d'une base de données évolutives au contenu géographique sous forme de faits et de règles de logique à laquelle des ontologies formelles sont appliquées. C'est alors qu'apparaît le défi de raisonner qualitativement à l'intérieur de telles bases de géocroyances.

1.2 Problématique

Lorsque nous pensons au mot "raisonnement", bien souvent, un des premiers "mots" que nous lui associons est le mot "logique", et ce, particulièrement dans les sciences, où généralement le raisonnement est de nature logique. Régulièrement, lorsque nous travaillons avec des informations spatiales, le travail accompli s'effectue de manière très formelle afin d'obtenir des résultats, à la suite à de raisonnements spatiaux, qui soient rigoureux. Cette réalité, a conduit des chercheurs tels qu'Edwards et Mostafavi à s'intéresser à des formalismes logiques qui respectent cette nécessité formelle de rigueur dans [MEJ04b]. Aussi, la structure de l'information spatiale est souvent complexe et les incohérences à l'intérieur des bases de géocroyances sont fréquemment distribuées de manière aléatoire. Il devient alors difficile d'imaginer que des approches numériques traditionnelles de recherche, programmées en langage C par exemple, soient plus simples ou avantageuses étant donné l'organisation imprévisible de l'information. Souvent, les problèmes de vérification de cohérence ontologique et de révision de croyances reviennent, sur le plan de la complexité, à chercher une aiguille dans une botte de foin. Ainsi, les approches logiques se prêtent bien à ce genre de problème puisqu'elles incluent à la fois un support représentationnel et des moteurs de raisonnement. Ces approches logiques reposent sur des méthodes par essais et erreurs auxquelles il est possible d'ajouter des heuristiques facilitant la recherche de solutions. En fait, en examinant certains problèmes spatiaux, il devient possible de réaliser que certaines approches logiques puissent venir simplifier ce que nous aurions souhaité pouvoir traiter directement d'abord numériquement. Mostafavi et Edwards ont montré qu'une approche logique pouvait très bien convenir pour traiter certains problèmes de raisonnement spatial.

La qualité des données est un enjeu majeur lorsque des interprétations sur l'espace physique doivent être tirées. Le maintien de la qualité des données ou de l'information spatiale est d'autant plus difficile lorsque l'information spatiale évolue au fil du temps. Pour maximiser la qualité des interprétations sur l'espace, il importe donc de s'assurer de la qualité de l'information spatiale à chaque instant. Ainsi, une façon de vérifier la qualité de l'information spatiale est de recourir à des contraites d'intégrité et de s'assurer que ces contraintes soient bien vérifiées. Aussi, la qualité des informations spatiales peut souvent être améliorée en les révisant afin de les rendre cohérentes entre elles, s'il y a des incohérences. Une telle opération s'effectue en ajoutant ou en retirant des informations spatiales d'un ensemble d'informations spatiales de façon stratégique de manière à tenir compte de la prévalence d'informations sur certaines autres.

Des travaux ont été menés en ce sens sur la modélisation des contraintes d'intégrité à l'intérieur de la BNDT par Normand et Bédard [Nor99]. Ces derniers ont développé un logiciel permettant de modéliser et de gérer les contraintes d'intégrité de la BNDT. À la fin de cet ouvrage, ils soulignent l'intérêt d'étudier les algorithmes de vérification de contraintes d'intégrité spatiale. Ceci laisse entendre que des progrès doivent encore être apportés au niveau du maintien de la cohérence des contraintes d'intégrité spatiale entre elles dans la BNDT. Il s'agit d'un aspect fondamental à considérer afin de pouvoir s'assurer que les résultats d'éventuelles analyses spatiales soient exacts.

Le raisonnement spatial qualitatif (*RSQ*) est très utile à la vérification de la cohérence ontologique de systèmes d'informations géographiques, ce qu'ont bien montré Mostafavi et Edwards dans [MEJ04b]. Il existe plusieurs méthodes pour la vérification de cohérence de contraintes d'intégrité, une opération de *RSQ*. Entre autres, des méthodes numériques de vérification peuvent être envisagées pour ce genre de tâche. Cependant, la programmation de telles méthodes peut devenir très complexe et fastidieuse et ces méthodes s'éloignent généralement quelque peu du fonctionnement du raisonnement humain. Se tourner vers une approche logique permet de mieux recréer le fonctionnement du raisonnement humain que des méthodes numériques qui sont souvent compliquées et difficiles à appliquer. D'ailleurs, une telle approche a été utilisée par Mostafavi et Edwards [MEJ04b].

Ces auteurs ont privilégié une approche ontologique utilisant la logique qui permet de générer un support représentationnel rigoureux pour les connaissances géographiques. L'emprise que nous avons sur les connaissances géographiques est renforcée grâce à une telle approche d'autant plus que le poids qui pourrait être associé à la dimension quantitative d'un tel problème est écarté. Aussi, traiter les connaissances géographiques à un niveau ontologique permet de mieux saisir la portée des incohérences qui pourraient exister au sein de celles-ci. Travailler à s'assurer de la cohérence des connaissances géographiques au niveau ontologique permet d'éviter des erreurs calculatoires et d'interprétation une fois rendu au niveau des données. Ainsi, il est avantageux de règler les problèmes de cohérence au niveau ontologique avant de raisonner à partir d'informations au niveau géographique.

Dans le travail de Mostafavi et Edwards [MEJ04b], il a été démontré qu'il était possible de déceler des incohérences dans de nombreuses situations ne contenant pas d'exceptions de nature spatiale. Dans la majorité des situations spatiales, il n'y a pas d'exceptions, ce qui signifie que de grands nombres de relations spatiales sont souvent bien définis et traduisent bien la réalité géographique sans présenter d'incohérences. Toutefois, il suffit, par exemple, de quelques oublis de la part des concepteurs de bases de données, comme des relations spatiales manquantes dans une base de données, pour que des incohérences engendrent des raisonnements faux réalisés par la suite sur cette dernière. Ainsi, l'approche des auteurs, reposant sur le *PROLOG*, un langage de programmation logique, permet de mettre en oeuvre des approches efficaces de raisonnement en géomatique dans certaines situations. Les auteurs en question sont parvenus à résoudre le problème de vérification de cohérence pour la plus grande partie des situations sujettes à se produire. Leur méthode de vérification se prête très bien à identifier des situations spatiales comportant des cas d'exceptions à l'origine d'incohérences mais ne permet pas de rétablir la cohérence d'informations spatiales. À la lumière de ces travaux, *le problème qui nous intéresse est à savoir s'il existe d'autres approches logiques qui pourraient nous permettre de vérifier la cohérence dans des cas où de l'information incomplète est utilisée* et *s'il serait possible d'implanter une approche logique pour rétablir la cohérence au sein de l'information spatiale*.

Or, dans le monde des représentations spatiales physiques, des situations d'exceptions, à l'origine d'incohérences, finissent très souvent par se présenter. Ces situations se présentent dans le problème de vérification de cohérence mais également dans les problèmes de fusion et de révision auxquels nous nous intéressons également. La révision est une opération essentielle pour être mesure d'assurer la cohérence d'informations spatiales et d'effectuer des raisonnements exacts à partir de celles-ci.

Dans plusieurs situations, dans le but de traiter des jeux de données multisources, il devient utile de réaliser des fusions d'ontologies, car c'est d'abord généralement à ce premier niveau qu'apparaissent des incohérences. Ceci, nous amène à examiner différentes stratégies d'intégration de données. La mise en commun de jeux de données spatiales implique souvent de s'assurer de la cohérence d'une telle mise en commun des sources d'informations géographiques. La fusion d'ontologies est une opération qui apparaît souvent nécessaire lors du traitement de l'information spatiale ou lorsque des ontologies sont mises en commun. Une telle réalité donne naissance à plusieurs questions. Que faire lorsque des jeux de données différents doivent être fusionnés ? Qu'arrive-t-il des règles qui régissent la cohérence propre à chacun de ces jeux de données ? Or, la fusion ou la mise en commun de différentes sources d'informations implique très souvent de la révision puisque les sources deviennent souvent incohérentes lorsqu'elles sont unies.

Les bases de géocroyances (BGC), que nous étudierons en détail dans ce mémoire, se révèlent des outils intéressants pour améliorer la modélisation des informations que nous possédons sur des territoires dans des contextes spatiaux en géomatique. Aujourd'hui, un des défis importants rencontrés dans les bases de géocroyances consiste en l'optimisation des tâches de raisonnement à l'intérieur de celles-ci lors de leur entretien et lors de la résolution de problèmes à caractère spatial. Cette répartition se complexifie au cours de l'ajout de nouvelles informations au sein de bases de géocroyances, ce qui a généralement pour conséquence d'affecter la cohérence interne de ces bases. L'utilisation d'implantations de moteurs *PROLOG* est très adéquate pour vérifier la cohérence d'une base de géocroyances à un instant donné ou dans un contexte où le raisonnement monotone suffit à représenter l'évolution de réalités spatiales avec des informations complètes. Toutefois, ce type d'implantation ne suffit pas à assurer le maintien de la cohérence au sein de bases de géocroyances qui se veulent évolutives et dont les informations contenues sont souvent incomplètes. Un tel type d'implantation rend difficile l'intégration de nouvelles informations et nécessite une reconsidération entière des croyances initiales contenues dans ces bases, ce qui s'avère un processus long et fastidieux.

Les limites qu'implique l'utilisation de la logique classique sont présentées par le groupe Léa Sombé dans *Raisonnements sur des informations incomplètes en intelligence artificielle* [dgLS89]. Elles se résument aux suivantes :

1. La logique classique permet difficilement de représenter les cas d'exception. Elle caractérise inadéquatement certains faits dans plusieurs situations concrètes. Par exemple, lorsque nous affirmons que "Les étudiants sont jeunes", nous signifions que "Généralement les étudiants sont jeunes" et non pas que "Tous les étudiants sont jeunes".
2. Exprimer toutes les exceptions liées à un contexte donné est souvent irréaliste.
3. Le différent n'est pas démontrable signifie qu'il est impossible d'inférer une conclusion à partir d'une formule logique telle que $Etu(x) \wedge \neg(x = Lea) \rightarrow Jeu(x)$. [dgLS89]. Si par exemple $x = Paul$, la formule se lit «Paul est étudiant et Paul n'est pas Léa, donc Paul est jeune». Or, il n'est pas possible de conclure que «Paul est jeune» puisque rien ne garantit que Paul n'est pas Léa et ainsi que Paul est différent de Léa. En effet, rien ne permet d'affirmer que nous n'avons pas $x = Paul = Lea$ dans les faits.
4. La représentation n'est pas modulaire puisque l'ajout de toute information implique de remettre en question la représentation de la connaissance.
5. Les symboles logiques "et" et "il existe" sont insuffisants pour représenter l'information qui comporte certaines exceptions. Des quantificateurs tels que "il existe au moins un" et "il existe au plus" sont nécessaires pour décrire certaines situations. La syntaxe de la logique classique est limitée pour la représentation de certains types d'information. Exemple : Une partie des routes sont des routes secondaires se traduit par " S'il y a une route alors il existe au moins une route secondaire".

Avec de telles limites, nous pouvons présager qu'il puisse devenir difficile de bien représenter des faits géographiques, traduisant des relations spatiales entre des entités telles que rencontrées entre des bâtiments et des routes par exemple, avec des implantations comme le *PROLOG*. Le *PROLOG* est un langage de programmation qui n'inclut pas la notion de modèle stable (Voir définition 2.21 pour plus de détails) et qui permet de mettre en oeuvre du raisonnement monotone et non monotone restreint dans des contextes spatiaux où l'information se veut souvent évolutive. Le raisonnement non monotone est un type de raisonnement logique qui permet de modifier les conclusions tirées antérieurement lors de l'ajout d'informations, ce que ne permet pas le raisonnement monotone.

Une des caractéristiques fondamentales de la logique classique est qu'elle ne permet pas le retrait de conclusions obtenues antérieurement lors de l'ajout de nouvelles informations. Ainsi, si une conclusion est tirée une fois, elle est toujours vraie dans un même programme logique. Seulement de nouveaux faits peuvent être ajoutés et de nouvelles conclusions déduites à l'intérieur d'un même programme logique. Or, dans un contexte comme celui qui nous intéresse, où de l'information spatiale évolutive est impliquée, cela pose un problème puisque des conclusions sur de l'information spatiale doivent parfois être retirées pour tenir compte de l'évolution de l'information spatiale. Le même type de situation est également rencontré lors de la fusion de différentes sources d'informations spatiales dans des bases de géocroyances.

Présentation du problème de recherche

Depuis plusieurs décennies, de nombreux chercheurs et professionnels oeuvrant dans différents domaines sont amenés à utiliser des données à caractère spatial afin de tirer des interprétations qualitatives et quantitatives sur l'espace physique. Un exemple de ces données est celui des données de la Base nationale de données topographiques canadiennes (BNDT). Dans le contexte de ce mémoire, nous choisissons le contexte d'application de la BNDT comme cas d'étude. Les principales particularités de la BNDT sont la complexité de son organisation et le nombre très élevé de relations spatiales qu'elle contient. Il s'ensuit que le maintien de la cohérence de l'organisation de la BNDT constitue une tâche complexe. Les approches de raisonnement que nous présentons dans le contexte de ce travail pourrait également s'appliquer à de nombreux autres contextes d'application que celui de la BNDT. En effet, les approches de raisonnement présentées pourraient très bien s'appliquer également à différentes branches de la géomatique, qu'il s'agisse de la télédétection, du positionnement par satellites ou encore de la photogrammétrie. Le problème de la cohérence de l'information spatiale est présent dans l'ensemble de ces différentes branches de la géomatique.

Cette complexité existe aussi à l'intérieur de chacune des données contenues dans la BNDT. Toutefois dans notre contexte, nous nous concentrons sur l'étude de l'organisation des données entre elles et non sur les données mêmes. Par exemple, nous nous intéresserons à la relation existant entre un chemin de fer et une route plutôt qu'à la relation d'intersection qui relie le chemin de fer national à l'autoroute 40. Ainsi, nous travaillerons sur l'ontologie de la BNDT, traduisant l'organisation des données plutôt que sur des instances de données.

La diversité des thèmes et des entités qui composent la BNDT rendent la définition de cette dernière une tâche nécessitant la plus grande minutie de la part des professionnels qui s'adonnent à ce travail. Bien que l'expertise de ces professionnels puisse permettre de restreindre le nombre d'incohérences entre les diverses composantes et spécifications ontologiques de la BNDT, un certain nombre d'incohérences subsiste au sein de celle-ci. Pour cette raison, le Centre d'information topographique de Sherbrooke (CITS) utilise une approche par échantillonnage pour la vérification de la cohérence ontologique de la BNDT. Ainsi, dans le cadre de ce mémoire, nous proposons une piste de solution afin d'offrir une approche complète pour la vérification de la cohérence et la révision de la BNDT, ce qui s'avère une motivation importante pour la réalisation de cette recherche.

Les relations entre les données décrites dans les tables de relations et cardinalités de la BNDT recèlent parfois des incohérences difficilement perceptibles. Il arrive fréquemment qu'il n'y ait pas de symétrie entre les relations spatiales décrites dans les tables. Par exemple, dans une table de relations de la BNDT, il sera indiqué qu'il existe une relation entre une route et un chemin de fer et dans une autre table, il ne sera pas spécifié qu'il existe une relation spatiale entre ce même chemin de fer et cette même route. Ainsi, ce type de problème de symétrie est fréquent et constitue un type de cas d'incohérence de la BNDT. De telles incohérences nuisent aux raisonnements spatiaux que nous pouvons effectuer à l'aide des informations contenues dans la BNDT. Or, afin d'être en mesure d'effectuer des raisonnements spatiaux qui soient pourvus de sens, il devient nécessaire, préalablement, de réviser les croyances sur l'ontologie de la BNDT afin de rétablir la cohérence au sein de celle-ci. Les croyances devant être révisées se présentent généralement sous forme de règles. L'approche de révision des croyances non stratifiées par la méthode des r-ensembles détaillée dans le chapitre 4 constitue celle que nous retenons dans le cadre de ce mémoire. Cette méthode s'avère efficace pour le traitement de l'information géographique qui nous intéresse.

Dans le but de réaliser une révision des croyances qui soit pertinente, il est important de définir la notion de qualité de l'information sur des données. Si des croyances sur des données doivent être supprimées ou modifiées, il faut être en mesure de bien identifier le degré de primauté d'une donnée sur une autre, et d'évaluer la qualité du processus de révision qui en découle. Ainsi, nous sommes amenés à poser un problème de recherche qui est lié à la primauté d'informations sur d'autres.

Les données du problème étudié correspondent aux descriptions des données et des relations contenues dans la BNDT recueillies sous forme de tables de relations et de cardinalités basées sur une modélisation entités-relations. À l'intérieur des tables de relations sont énumérées les différentes entités qui composent la BNDT auxquelles sont rattachées des relations spatiales lesquelles sont décrites à l'intérieur de ces tables. Chaque ligne de ces tables constitue un fait ou un ensemble de faits selon le nombre de codes impliqués dans les relations qu'elle décrit. Les données sont définies sous forme de thèmes, d'entités, de combinaisons et de codes. Les modélisations entités-relation conviennent bien pour structurer l'information géographique et pour avoir une bonne emprise sur les détails de son organisation puisque chaque ligne permet de décrire une loi que doit respecter une entité géographique. Des difficultés surviennent toutefois lorsque vient le moment de vérifier la cohérence de l'ensemble de ces lois entre elles.

Voici en quelques lignes ce à quoi se résume l'organisation de la BNDT :

À chaque code présenté dans les tables de relations de la BNDT correspond trois éléments soient : une entité, une combinaison d'entités et une cardinalité. Les relations possibles entre un code et d'autres codes sont présentées à l'intérieur de tables des relations et cardinalités de partage, de connexion ou d'adjacence et de superposition.

Il existe 14 thèmes (hydrographie, réseau routier, etc.) qui comprennent chacun plusieurs entités. Chaque entité possède à son tour plusieurs attributs. Les attributs se rattachent aux entités suivant certaines combinaisons qui sont permises. Les combinaisons permises sont établies selon les recommandations des responsables de la BNDT qui s'appuient sur leur expérience cartographique.

Dans le cadre de ce mémoire, nous nous concentrerons sur les relations de connexions décrites dans les champs des tables de relations et de cardinalités de connexion. Ceci constituera le domaine du problème de la BNDT que nous abordons.

En *PROLOG*, les seules clauses permises pour la caractérisation d'un problème sont des clauses de Horn, des clauses comportant au plus un littéral positif. Par exemple, $\neg a \vee \neg b \vee \neg c \vee d$ est une clause de Horn. Lorsqu'une clause de Horn est exprimée sous forme de règle, une seule conclusion possible peut être tirée. Or, lorsque des informations sont incertaines ou incomplètes, il arrive que plusieurs conclusions soient possibles. Ainsi, lorsque plusieurs conclusions sont possibles, il est nécessaire d'utiliser des clauses qui ne sont pas des clauses de Horn afin de représenter ce type de situation. Lorsque nous rencontrons des situations où les clauses de Horn ne suffisent pas pour représenter un problème, le *PROLOG* devient un langage dont les possibilités de représentation sont trop limitées. Une alternative intéressante devient celle de recourir à l'*ASP* pour caractériser notre problème. L'*ASP* permet d'avoir recours à des clauses de quelconque nature pour des fins de représentations de situations, ce, contrairement au *PROLOG*. Ainsi, dans ce mémoire nous verrons, entre autres, que les clauses de Horn sont insuffisantes pour représenter certaines relations spatiales.

1.3 Questions de recherche et objectifs de recherche

Dans cette section, nous formulons les questions et les objectifs en lien avec notre problème de recherche.

1.3.1 Questions de recherche

À la lumière de la problématique précédente, nous formulons les questions de recherche suivantes :

1. Comment s'opère le raisonnement en logique classique et dans le contexte du raisonnement par défaut ?
2. Est-ce que l'ingénierie des connaissances peut constituer un support efficace pour la représentation d'ontologies géospatiales ?
3. Comment se met en oeuvre le raisonnement non monotone en *PROLOG* et en *ASP* ?
4. Quel est l'apport du concept de modèle stable existant en *ASP* mais absent en *PROLOG* pour les opérations de vérification de cohérence ontologique et de révision des géocroyances lorsque des informations incomplètes ou comportant des exceptions doivent être traitées ?
5. En quoi les possibilités liées à l'utilisation des moteurs de raisonnement *PROLOG* ou *Smodels* pour l'opération de vérification de cohérence ontologique diffèrent-elles ?
6. En quoi les possibilités liées à l'utilisation des moteurs de raisonnement *PROLOG* et *Smodels* pour l'opération de révision des géocroyances diffèrent-elles ?

1.3.2 Objectifs

L'objectif principal du présent mémoire est d'implanter et de comparer une approche de raisonnement qualitatif non monotone en *PROLOG* avec une approche de raisonnement non monotone en *ASP* dans un contexte géospatial. Les objectifs spécifiques qui en découlent sont les suivants :

1. Explorer la logique classique et le raisonnement par défaut.
2. Définir et créer une base de géocroyances en formalisant des ontologies et en utilisant une approche d'ingénierie des connaissances adaptée au contexte spatial.
3. Expérimenter le raisonnement non monotone en *PROLOG* et en *ASP* dans une base de géocroyances.
4. Évaluer l'apport du concept de modèle stable en *ASP*, absent en *PROLOG*, pour les opérations de vérification de cohérence ontologique et de révision des géocroyances impliquant des informations incomplètes ou comportant des exceptions.
5. Évaluer et comparer les possibilités liées à l'utilisation des moteurs de raisonnement *PROLOG* et *Smodels* pour l'opération de vérification de cohérence ontologique.
6. Évaluer et comparer les possibilités liées à l'utilisation des moteurs de raisonnement *PROLOG* et *Smodels* pour l'opération de révision des géocroyances.

1.4 Méthodologie

Cette recherche est accomplie dans le cadre d'une Maîtrise en géomatique réalisée à l'Université Laval sous la direction de messieurs Geoffrey Edwards et Mir Abolfazl Mostafavi. Elle a également été l'occasion d'établir une collaboration avec madame Odile Papini, professeure à l'Université de la Méditerranée à Marseille et Éric Würbel, professeur à l'Université du Sud - Toulon - Var.

L'étude de la logique effectuée à l'intérieur de cette collaboration nous conduit à croire que certaines implantations d'approches de raisonnement non monotones peuvent constituer des outils bien utiles pour raisonner dans des contextes d'intégration de nouvelles informations et sur des informations incomplètes contenues dans les bases de géocroyances. Certaines implantations comme *Smodels* permettent de mieux traiter des cas d'exception dans une base de géocroyances et de traiter efficacement l'information incomplète. De telles implantations permettent également de réaliser des fusions d'ontologies et des révisions de géocroyances. Nous verrons en détail dans le quatrième chapitre de ce mémoire comment en plus d'une vérification de cohérence ontologique, une révision de géocroyances peut être accomplie.

Un des intérêts principaux que nous entrevoyons, lié à l'utilisation d'une approche non monotone, est la possibilité de mieux épouser la réalité lors de raisonnements spatiaux tout en proposant de nouvelles représentations de connaissances spatiales. Dans le troisième chapitre de ce mémoire portant sur les bases de géocroyances, nous verrons ce en quoi consistent ces nouveautés au niveau des possibilités de représentation et dans le quatrième chapitre ce qui en est pour le raisonnement spatial.

Par exemple, la logique des défauts, une logique non monotone, permet le retrait de conclusions précédemment déduites, ce qui explique notre intérêt pour l'exploration d'implantations de moteurs de raisonnement non monotone. Ceci nous amène également à nous intéresser à des approches de raisonnement non monotone qui permettent de tels retraits de conclusions.

Si de telles approches se révélaient concluantes, le temps de raisonnement à l'intérieur de bases de géocroyances pourrait être considérablement réduit dans certaines situations lorsque des heuristiques de raisonnement appropriées seraient appliquées à l'intérieur d'implantations d'approches non monotones. Aussi, cette approche pourrait permettre d'accélérer le processus de validation de la cohérence ainsi que celui de révision des croyances au sein d'une base de géocroyances. Également, l'utilisation d'une approche de raisonnement non monotone serait plus pertinente qu'une rapproche de raisonnement classique puisque qu'elle épouse mieux la façon dont l'humain raisonne naturellement. Cela constituerait un pas vers le développement d'outils géomatiques dont la structure de fonctionnement s'apparente à celle du raisonnement humain. Dans ce qui suit, nous présentons les dimensions importantes du problème de la cohérence de la BNDT, soit le problème principal auquel nous nous attarderons à résoudre dans le cadre de ce mémoire.

La principale logique non monotone dont différentes implantations présentent un intérêt pour nous est la logique des défauts introduite par Reiter [Rei87]. Cette logique possède la qualité de rendre possible le traitement d'exceptions dans son mécanisme d'inférence permettant de tirer des conclusions d'ordre logique. Elle permet également l'ajout et le retrait de faits et de conclusions sans compromettre la cohérence interne au sein d'une base de géocroyances. Le raisonnement par défaut permet de réaliser des déductions en l'absence de certaines informations. La logique des défauts constitue une logique non monotone utilisée afin de formaliser le raisonnement par défaut. Elle est pertinente lorsque nous tâchons de décrire des situations telles que : " Normalement ou typiquement, les A sont des B (avec des possibilités d'exceptions) " Léa Sombé [dgLS89]. Elle est bien adaptée au traitement et à la représentation de règles avec exceptions. Elle présente un intérêt particulier dans le cas qui nous intéresse, car la sémantique des modèles stables pour les programmes logiques normaux, lesquels ont la particularité de ne pas contenir de symbole de négation classique "$\neg$", en constitue une réduction. Il s'ensuit qu'il devient intéressant d'étudier *Smodels*, une mise en oeuvre utilisant la programmation logique avec sémantique de modèles stables (ou ensembles-réponses), qui permet de mettre en application des approches de raisonnement non monotone comme par exemple le raisonnement par défaut. La sémantique des modèles stables définit des ensembles de modèles, constituant des ensembles solution, validant un programme logique. Nous détaillerons les caractéristiques de *Smodels* dans le cadre théorique du présent mémoire. De plus, *Smodels* nous servira à simuler le raisonnement non monotone à l'intérieur d'une base de géocroyances. Nous étudierons également un moteur de raisonnement *PROLOG* qui permet également d'effectuer du raisonnement par défaut et non monotone mais qui est plus restreint que le moteur de raisonnement *Smodels* dans ses possibilités dans le cas du raisonnement non monotone.

Dans ce qui suit, nous présenterons les différentes étapes que nous avons dû franchir pour parvenir à atteindre les objectifs fixés dans le cadre de ce mémoire.

La démarche que nous avons suivie a comporté deux principales phases qui sont les suivantes :

Phase 1 : Exploration des caractéristiques et limites de différentes logiques dans un contexte de programmation pour différentes situations de raisonnement spatial.

1) Nous avons exploré les limites d'approches logiques monotones ainsi que d'approches non monotones. Nous avons établi clairement les qualités et faiblesses de ces approches en lien avec leur contexte d'utilisation. Deux sessions de travail d'un peu plus d'une semaine ont été organisées avec Odile Papini afin de bien saisir les bases de ces types d'approches logiques. Nous avons examiné comment les approches non monotones s'appliquent au problème des bases de croyances. Cette étape s'est révélée un moment opportun pour procéder à l'identification de contextes spatiaux particuliers et pour l'application de la méthode de raisonnement développée par Odile Papini. Nous avons étudié comment le recours à des approches non monotones devient essentiel pour traiter des cas d'exception, des cas d'informations incomplètes et pour traiter des cas d'ajout de nouvelles informations qui ont des incidences sur la cohérence globale de BGC. Ainsi, nous avons expérimenté le raisonnement logique non monotone en *PROLOG* et en *ASP* dans une base de géocroyances.

2) Nous nous sommes familiarisés avec le raisonnement non monotone simulé à l'aide de moteurs de raisonnement de logique non monotone.

Phase 2 : Application et comparaison à des exemples de relations spatiales incohérentes issues de la BNDT de deux méthodes de raisonnement non monotone, soit dans un premier temps en utilisant une méthode reposant sur le *PROLOG*, et dans un deuxième temps en recourant à une méthode de raisonnement faisant intervenir le concept de modèles stables, un concept qui permet d'identifier efficacement et stratégiquement des incohérences contenues dans un ensemble d'informations, en utilisant l'*ASP*.

1. Nous avons choisi un échantillon de la BNDT pour former une base de géocroyances qui constitue une formalisation de l'ontologie de la BNDT réalisée à l'aide d'un formalisme comme le *PROLOG*. Cet échantillon a été créé par Mostafavi [MEJ04b]. Pour la réalisation de cette étape, nous nous sommes appuyés sur les travaux de Mostafavi [MEJ04b] réalisés dans le cadre de son post-doctorat.
2. Nous avons construit une base de géocroyances en utilisant une approche d'ingénierie des connaissances suggérée par Gruber [Gru93b].
3. Nous avons évalué l'apport du concept de modèle stable en *ASP*, absent en *PROLOG*, pour les opérations de vérification de cohérence ontologique et de révision des géocroyances impliquant des informations incomplètes ou comportant des exceptions.
4. Nous avons évalué et comparé les possibilités de raisonnement logique liées à l'utilisation des moteurs de raisonnement *PROLOG* et *Smodels* pour l'opération de vérification de cohérence ontologique.
5. Nous avons appliqué une méthode de révision de croyances à l'intérieur d'une base de géocroyances.
6. Nous avons évalué et comparé les possibilités de raisonnement logique liées à l'utilisation des moteurs de raisonnement *PROLOG* et *Smodels* pour l'opération de révision des géocroyances.

1.5 Présentation du mémoire

Dans le chapitre 2 *Cadre théorique* de ce mémoire, nous présentons les bases théoriques sur lesquelles nous nous appuyons pour réaliser notre recherche. Pour commencer, nous abordons la théorie sur les ontologies, d'abord sous l'angle de l'informatique et ensuite sous celui de la géomatique. Nous poursuivons avec les fondements de la logique, en discutant de différentes logiques pertinentes pour notre problème de recherche. Nous définissons ce que nous entendons par base de géocroyances. Nous présentons la notion de raisonnement en logique et les différentes formes que ce raisonnement peut prendre soit monotone, soit non monotone. Ceci nous amène à parler de raisonnement spatial, ce qui est notre principal intérêt en géomatique dans le cadre de ce mémoire. La cohérence n'étant pas toujours effective dans les bases de géocroyances, nous présentons une méthode de révision des croyances afin de la restaurer. Étant donné qu'il existe souvent une confusion entre les notions de mise à jour et de révision, nous présentons et expliquons brièvement les distinctions existant entre ces deux notions dans le contexte de cette recherche. Finalement, nous discutons d'implantations existantes d'approches logiques monotones et non monotones.

Dans le chapitre 3 *Bases de géocroyances*, nous exposons les fondements des bases de géocroyances. Ce chapitre est aussi l'occasion de définir une base de géocroyances pour un contexte spatial particulier qui sera étudié. Ainsi, nous décrivons les composantes d'une base de géocroyances. Nous discutons de différentes représentations possibles de bases de géocroyances, en nous appuyant sur les travaux de Mostafavi 2004 [MEJ04b]. Nous choisissons le cas de la BNDT en tant que cas d'étude et nous décrivons son ontologie de même que l'intérêt qu'elle comporte dans le contexte de la géomatique.

Le chapitre 4 fait l'objet de l'étude de la *Traduction et vérification de la cohérence d'une ontologie dans une base de géocroyances*. Nous définissons le raisonnement spatial appliqué au contexte de la BNDT. Nous abordons les notions de raisonnements monotone et non monotone au sein de la BNDT. Dans ce chapitre, nous traduisons un contexte spatial particulier propre à une base de géocroyances issue de la BNDT, d'abord en *PROLOG* et ensuite en *Answer Set Programming* (*ASP*). Nous comparons ensuite les possibilités de raisonnement spatial liées à l'utilisation de ces deux langages. Nous analysons les résultats obtenus. Nous étudions ce qu'impliquerait d'effectuer une révision des croyances. Enfin, nous discutons de la complexité des traductions et de l'ontologie de la BNDT, en *PROLOG* et en programmation par ensemble de réponses (*ASP*).

Le chapitre 5 est l'occasion d'effectuer une synthèse des principaux résultats obtenus dans le chapitre 4 et de tirer des conclusions sur l'ensemble de la recherche réalisée dans le cadre de ce mémoire. Enfin, nous faisons part de travaux à entreprendre pour améliorer l'efficacité du raisonnement à l'intérieur de bases de géocroyances.

Chapitre 2

Cadre théorique

Dans le présent chapitre, nous exposons les principales bases théoriques sur lesquelles nous nous appuyons dans le cadre de ce mémoire. Ceci nous a mené à traiter de la théorie sur les ontologies ainsi que de notions de logique. Ces bases théoriques serviront principalement à la définition de bases de géocroyances étendues dans le chapitre 3 et pour mener différentes opérations de raisonnement spatial dont la fusion et la révision dans une base de géocroyances dans le chapitre 4.

2.1 Les ontologies et la géomatique

L'établissement des fondements ontologiques en géomatique est assez récent [BE01] comme nous le verrons dans la section portant sur les travaux de Bittner et Edwards. Ceci nous conduit à nous demander d'où provient cette nécessité de recourir à des ontologies dans un tel domaine ? La géomatique se situe au confluent d'une panoplie de domaines parmi lesquels se trouvent l'arpentage, la géographie, la télédétection, la gestion de bases de données, l'informatique et de nombreux autres. Ainsi, puisque la géomatique est une science pluridisciplinaire, il importe de bien saisir les fondements des différents domaines qui lui sont rattachés.

En géomatique, la notion de territoire est généralement associée au domaine géographique. Or, "Comprendre le domaine géographique, c'est tout d'abord comprendre son ontologie, c'est-à-dire, la nature des objets géographiques, les relations entre eux, la structure de base de l'espace lui-même et la nature du lieu et de la référence spatiale" [BE01]. Un tel fait nous conduit à réfléchir sur les fondements ontologiques en géomatique et à nous interroger à savoir quels pourraient être les éléments que devrait englober une ontologie dans le contexte de la géomatique.

Les travaux de Bittner et Edwards

D'après Bittner et Edwards (2001) dans l'article "Towards an ontology for geomatics" [BE01] une ontologie en géomatique devrait comporter les éléments et caractéristiques suivants :

– La définition d'une ontologie devrait intégrer la notion d'objets. Pour pouvoir accomplir des tâches de nature géomatique, il est fondamental de pouvoir reconnaître les entités avec lesquelles nous travaillons, et conséquemment, de pouvoir les désigner et en reconnaître les caractéristiques.

– Une ontologie devrait inclure la notion d'ambiguïté, d'indétermination et de référence. Dans le contexte de la géomatique, ces notions reviennent fréquemment. Les indéterminations spatiales découlent de l'existence d'ambiguïtés au sein des propriétés qui caractérisent les objets. L'indétermination vient alors affecter la référence spatiale propre aux objets géomatiques. Ainsi, une ambiguïté naît au niveau de la référence spatiale, ce qui entraîne des conséquences importantes dans ce domaine.

– Une ontologie devrait appuyer l'intégration des observations auxquelles peuvent être rattachées des mesures et des analyses. Cette dimension est très importante en ce sens qu'elle se situe à l'origine d'éventuelles possibilités d'interopérabilité. Un des objectifs visés en définissant des ontologies est de rendre possible la communication entre différents systèmes, et ce, indépendamment de leurs technologies respectives. Il est fondamental que les systèmes préservent leur autonomie et leur flexibilité. Pour cette raison, le choix d'une ontologie devra être effectué judicieusement. Ainsi, l'ensemble de ces éléments devra être considéré lors de la définition d'une base de géocroyances (BGC).

Toujours selon Bittner (2001), les ontologies en géomatique devraient contenir plusieurs types de relations. Entre autres, des relations tout-parties devraient être présentes. Les auteurs entendent par là des relations entre des entités et les parties dont elles sont composées. Il devrait également s'y trouver des relations de type lieu-référence. Il s'agit de relations entre le lieu que nous désignons par un nom et sa référence spatiale correspondante. De même, des relations caractérisant la configuration spatiale devraient être établies. Ces relations peuvent être des relations existant au sein de topologies, comme la connectivité et des métriques.

La géomatique doit avoir recours à quatre niveaux différents d'analyse afin de permettre de définir des ontologies qui puissent être pertinentes. Ces niveaux sont :

– Le niveau ontologique formel
– Le niveau épistémique
– Le niveau du contexte d'application
– Le niveau de calculabilité

Ces quatre niveaux devront être étudiés en profondeur lorsque viendra le moment de définir une base de géocroyances adaptée au raisonnement spatial, ce qui sera réalisé un peu plus loin dans ce mémoire.

Le niveau ontologique formel

À ce niveau, les types d'objets et de phénomènes avec lesquels nous travaillons sont définis. Aussi, les aspects structuraux sont-ils définis. Il peut s'agir de la composition topologique, des relations entre les objets, de leur position dans l'espace et du temps, ou encore des relations entre les objets et leur environnement. Le niveau d'ambiguïté est défini de même que celui d'indétermination et de référence. La définition d'une ontologie formelle procure un outil, un vocabulaire, et des lois qui régissent les objets et phénomènes spatiaux. Pour bien représenter la réalité spatiale en géomatique, il est intéressant d'introduire des relations SPAN/SNAP [GS04]. SPAN est une théorie ontologique qui soutient que des entités se déploient à travers le temps. Ainsi, une ontologie SPAN s'obtient en représentant la réalité constituée d'entités qui se déploient à travers des intervalles de temps déterminés dans certains domaines de la réalité et à des niveaux de granularité donnés. Ces entités peuvent être des entités de nature géographiques. Cette théorie permet de composer avec la dimension temporelle qu'impose la majorité des projets de géomatique. Une ontologie SNAP s'obtient à travers la représentation d'entités durables à un temps donné. Ce type d'ontologie est comparable à un cliché. Les entités SNAP sont décomposables et cumulatives. Elles ne possèdent pas de partie temporelle. La somme de diverses entités SNAP donne naissance à une nouvelle entité SNAP. Chaque ontologie SNAP est indexée en lien avec un moment précis dans le temps

Ainsi, lorsqu'il s'agit d'espaces géographiques, il est primordial de bien saisir les relations entre l'ontologie d'un espace géographique et les conceptualisations relevant de domaines spécifiques. De telles conceptualisations de l'ontologie doivent bien épouser la réalité. Elles dépendent des besoins et de la perception de l'espace du concepteur de l'ontologie.

Le niveau épistémique

Le niveau épistémique correspond au niveau auquel nous tâchons d'établir ce qu'il est possible de connaître en termes d'instances d'objets et de phénomènes issus des observations et prises de mesures. À ce niveau sont explorés les liens entre la nature des observations des objets spatiaux, les relations entre l'ambiguïté présente dans la définition d'objets, l'indétermination existant dans la position, ainsi que l'incertitude sur la vérité de connaissance gagnée par l'observation et la prise de mesure. Ce type d'analyse permet de donner un sens à des concepts comme ceux de résolution, de tolérance et d'erreurs. Les théories aux niveaux épistémique et ontologique formel procurent des fondements pour la représentation spatiale dans un contexte de géomatique.

Le niveau du contexte d'application

Les spécialistes d'applications pour un contexte donné doivent fournir des informations détaillées sur les objets et les phénomènes propres à des domaines d'application spécifiques. Il peut s'agir de régions polluées, de régions défavorisées ou de multiples autres sortes de régions. Ces spécialistes doivent pouvoir indiquer ce qu'il y a lieu d'observer et de mesurer. Il devient alors possible de collecter l'information nécessaire afin de procéder à des analyses spatiales. Or, l'analyse spatiale peut être simplifiée par les connaissances acquises aux niveaux ontologiques et épistémologiques concernant la nature et les propriétés des objets. Au niveau des applications, les ontologies sont généralement organisées sous forme de taxonomie. Un exemple intéressant est celui de l'ontologie de la BNDT. Celle-ci permet à des spécialistes d'applications spatiales de s'assurer de la cohérence de l'information spatiale ou d'estimer le niveau d'incohérence avant d'effectuer des analyses spatiales. Ensuite, ces spécialistes sont en mesure d'informer les décideurs liés à des projets sur les conclusions qui sont possibles de tirer à la lumière des analyses spatiales et de la qualité de la cohérence des informations spatiales.

Le niveau de calculabilité

Le développement d'outils d'analyse spatiale implique souvent un grand nombre de méthodes de calculs qui sont implantées à l'intérieur d'algorithmes. Ces outils d'analyse spatiale jouent un rôle très important car ils réduisent la complexité des processus de prise de décisions, ainsi, ils se doivent donc d'être adéquats. Dans le domaine de l'intelligence artificielle, les chercheurs essaient de modéliser le processus décisionnel humain en recourant à des approches comme le raisonnement par cas. Ces chercheurs tentent également d'extraire automatiquement des lois qui régissent les objets à partir des connaissances qui sont disponibles sur les observations (datamining). Or, tous ces processus dépendent grandement des ontologies définies préalablement. Ces ontologies viennent affecter la calculabilité des algorithmes de raisonnement humain impliquant, entre autres, des analyses spatiales.

En résumé, pour ces auteurs, les ontologies possèdent plusieurs qualités. Une qualité importante des ontologies est de bien traduire la réalité en tenant compte de la perception humaine de celle-ci. Il est bien important lors de l'élaboration d'ontologies de conserver en mémoire l'idée que lorsque nous élaborons des ontologies dans le contexte de la géomatique, l'organisation des éléments au sein de ces dernières aura un impact autant au niveau de la façon dont nous concevons les objets spatiaux et leurs relations que sur les algorithmes qui serviront à raisonner et à réaliser des analyses spatiales sur l'espace physique.

Nous allons maintenant examiner les travaux de Smith et Mark en nous attardant plus particulièrement à leur contribution à la réflexion sur le choix des termes pour nommer les classes et les superclasses dans les ontologies dans le contexte géographique.

Les travaux de Smith et Mark

Smith et Mark (2001) [SM01], dans le but de définir des ontologies géographiques, ont réalisé une série d'expériences visant à saisir comment des sujets non-spécialistes en géographie se représentaient les phénomènes géospatiaux. Une telle expérience permet de saisir comment des sujets conçoivent leur environnement spatial de façon naturelle. Les chercheurs ont demandé à 263 sujets de procéder à la lecture d'énoncés descriptifs et de donner des exemples de classes d'éléments géographiques. Le résultat de l'expérience a conduit les auteurs à établir une ontologie de classes géographiques. Cette démarche leur a permis de réaliser une compilation des principaux concepts et classes géospatiaux que partagent la plupart des humains.

L'étude de la façon dont les sujets non-spécialistes des phénomènes géospatiaux conceptualisent les différents domaines de la réalité rend l'information contenue dans les systèmes d'information plus facilement utilisable dans différents contextes. Cette forme d'étude se veut donc très pertinente.

Les expériences menées par Smith et Mark offrent un premier aperçu des principales ontologies géospatiales partagées par les utilisateurs de systèmes d'information géographique (SIG). Elles montrent également que le mur qui sépare la langue d'une ontologie n'est pas facile à franchir. Cela a été vérifié dans le cas de la géographie et laisse croire que cela pourrait aussi être applicable de manière générale.

Tels que le rapporte Smith et Mark, les philosophes ontologiques sont conscients depuis longtemps de la problématique que pose la terminologie des ontologies. Les auteurs de l'article montrent bien comment le choix du nom d'une superclasse propre à une ontologie aura une influence sur le choix des classes la composant. Une ontologie diffère par sa structure selon le nom choisi pour désigner sa superclasse. Dans leurs travaux de recherche, les auteurs sont parvenus à définir une ontologie géographique qui contient les cinq classes suivantes : montagne, rivière, lac, océan et mer. Il semble que pour les sujets non-spécialistes ces chercheurs soient arrivés à la conclusion que ces principales composantes de l'environnement physique constituent les meilleurs exemples de phénomènes géographiques pouvant servir de repère chez l'humain, à prendre en compte pour la conception de cartes. Ces travaux montrent que le choix de superclasses et de classes pourra avoir un impact sur la capacité des sujets non-spécialistes à repérer les éléments qu'ils recherchent dans des cartes. Ainsi, cette recherche met en évidence le caractère fondamental de la définition d'une ontologie qui tienne compte de la spécificité des utilisateurs auxquels d'éventuelles cartes sont destinées. Ces résultats nous invitent à bien réfléchir sur la terminologie que nous utiliserons lors de la définition de l'ontologie à laquelle nous recourrons pour l'élaboration d'une base de géocroyances.

Des travaux plus récents, soit ceux de Mostafavi, Edwards et Jeansoulin (2004) [MEJ04b], ont permis de pousser plus loin la réflexion sur les ontologies cette fois en tentant, en plus de s'assurer d'une définition d'ontologies de qualité, de procéder à la vérification de la cohérence d'ontologies.

Les travaux de Mostafavi, Edwards et Jeansoulin

Mostafavi, Edwards et Jeansoulin (2004) [MEJ04b] proposent dans "An ontology-based method for quality assesment of spatial data bases" une approche ontologique pour le maintien de la qualité des bases de données spatiales. Leur approche ontologique possède deux niveaux. D'abord, il y a le niveau ontologique auquel la cohérence interne de la base de données spatiales est étudiée et, ensuite, il y a le niveau de données auquel des objets et leurs relations en liens avec leur ontologie sont étudiés. Ils ont utilisé la BNDT, soit la Base nationale de données topographiques du Canada comme cas d'étude. Dans ces travaux Mostafavi et *al* proposent de représenter l'ontologie de la BNDT en *PROLOG*, ils formulent des règles de vérification de cohérence dans ce même langage et parviennent à déceler un bon nombre d'incohérences logiques dans la BNDT en raisonnant à partir de ces règles. Ces travaux nous servent de point de départ pour la rédaction de ce mémoire.

Nous appuyant sur l'ensemble de ces travaux, nous pouvons conclure qu'il n'est pas aisé de définir des ontologies. Néanmoins, ces dernières apportent un appui de taille au raisonnement logique en tant que mode de représentation des croyances.

Définir des ontologies implique de représenter des entités sous forme de classes. Une des caractéristiques intéressantes des ontologies est la structure modulaire qu'elles engendrent. Les organisations modulaires ont l'avantage de pouvoir être réutilisées dans divers contextes lorsqu'elles sont jugées pertinentes. Pour les rendre utilisables, il faut alors instancier la classe ou le module dont nous voulons nous servir.

Souvent pour traduire des situations et raisonner sur ces dernières, nous recourons à la logique classique. Or, lors de l'accomplissement de raisonnements s'appuyant sur une telle logique, nous nous retrouvons dans des situations où la sélection de clauses est conditionnée par le résultat des résolutions précédentes liées au processus de raisonnement.

Dans la section qui suit, nous présentons quelques éléments de base de la logique.

2.2 Éléments de logique classique

Dans cette section, nous commencerons par présenter quelques faits marquants dans l'histoire de la logique. Ensuite, nous nous attarderons à quelques notions élémentaires de logique qui présentent un intérêt certain pour mener à bien à la fois les opérations de définition et de vérification. Ceci nous conduira à parler de la logique formelle, de la logique propositionnelle, la logique des prédicats et la logique possibiliste. Pour ces deux dernières logiques, nous présenterons leurs langages, sémantiques, syntaxes et axiomatiques propres.

La logique utilisée en informatique se veut une logique formelle contrairement à la théorie de la connaissance ou encore à l'épistémologie, et elle fournit une théorie de l'inférence valide au sens formel tel que le laisse entendre Roger Martin [Mar58] dans le paragraphe qui suit. En logique formelle, toute inférence reconnue comme vraie, par un sujet pourvu de sens et dont l'expérience peut être jugée suffisante, est valide, dans la mesure où celui-ci considère vraies ses prémisses.

Tel que le résume Roger Martin [Mar58], si en apportant des modifications, à l'intérieur de limites préétablies, aux éléments d'une phrase qui traduit une inférence, cette dernière demeure valide, nous conclurons que l'inférence est valide. Nous dirons que la phrase «Si il y a une tempête de neige, alors les routes sont dangereuses » constitue une inférence valide mais qu'elle ne peut pas être considérée comme formellement valide puisqu'aucun fait dans la phrase ne nous informe si il y a une tempête de neige ou non. Cette phrase ne contient qu'une règle d'inférence. Toutefois, nous affirmerons que «Si il y a une tempête de neige, alors les routes sont dangereuses ; effectivement, aujourd'hui il y a une tempête de neige ; donc ceci laisse entendre que les routes sont dangereuses aujourd'hui.» est une inférence formellement valide parce qu'elle contient une information factuelle qui permet de réaliser l'inférence. La manière la plus élémentaire d'établir si l'inférence étudiée se révèle être une inférence formelle consiste à identifier les éléments d'un texte qui peuvent être substitués sans affecter le sens de l'inférence qui est définie. Ainsi, dans le dernier exemple d'inférence mentionné, nous pouvons susbtituer certaines parties de la phrase par des variables propositionnelles comme a et b. Nous obtenons d'une telle manière «Si a, alors b, effectivement a, donc b.».

Toujours suivant Martin [Mar58], les premiers usages systématiques de variables remontent à la syllogistique d'Aristote. Un des premiers exemples fut l'utilisation des variables « homme » et « mortel » dans les syllogismes d'Aristote comme dans «tout homme est mortel ; or Socrate est un homme ; donc Socrate est mortel». Il apparaît que chaque logique formelle possède un caractère symbolique. Ceci conduisit à l'apparition des premiers prédicats. La logique d'aujourd'hui est munie d'un grand nombre de variables, qu'il s'agisse de variables propositionnelles, de prédicats, de fonctions, de relations ou encore d'individus.

Avant d'entreprendre la section sur la logique propositionnelle, nous résumons les distinctions entre la sémantique et la syntaxe. Ainsi, la syntaxe d'une logique réfère aux règles de grammaire propres à un langage propositionnel ou prédicatif (variables propositionnelles et connecteurs) alors que la sémantique d'une logique correspond aux valeurs de vérité vraie ou fausse de l'interprétation de formules logiques.

2.2.1 La logique propositionnelle

La logique propositionnelle correspond à la partie de la logique qui traite des propositions. Une proposition est une affirmation munie d'un sens. Par exemple, "Il fait beau". À des propositions sont entachées des valeurs de vérité. L'objectif fondamental de la logique propositionnelle est de rendre possible le raisonnement à partir de propositions, ce qui donne naissance à ce qui est appelé le "calcul propositionnel". Dans ce type de calcul, les propositions sont traitées comme des variables à partir desquelles des opérations sont réalisées.

Dans les lignes qui suivent, nous présentons quelques éléments de logique propositionnelle afin de résumer les bases de cette logique. Nous nous appuyons sur les annexes A et B du livre *Raisonnements sur l'espace et le temps : des modèles aux applications* de Papini [GLP07] portant respectivement sur la logique propositionnelle et sur la logique des prédicats à paraître au courant de l'année 2007 chez Hermes. Nous reprenons également les notations utilisées par ces auteurs dans la présente section sur la logique propositionnelle et la suivante sur la logique des prédicats.

La logique propositionnelle est munie d'une syntaxe, d'un langage, d'une axiomatique et d'une sémantique. Dans ce qui suit, nous présentons ces différents aspects de la logique propositionnelle.

1. Aspects syntaxiques de la logique propositionnelle

La syntaxe de la logique propositionnelle permet de déterminer quelles sont les formules bien formées à l'intérieur du cadre appartenant à la logique propositionnelle. Cette première fait abstraction de la signification des propositions et de leurs valeurs de vérité. Un calcul logique constitue un ensemble de formules propositionnelles appelées théorèmes qui est déterminé par un ensemble de formules propositionnelles appelées axiomes et de règles d'inférence. Ce calcul logique a pour principal utilité de rendre possible le raisonnement à l'intérieur d'un cadre logique.

Le langage de la logique propositionnelle

Le langage de la logique propositionnelle $\mathcal{L}$ est élaboré à partir d'un ensemble infini dénombrable de variables propositionnelles aussi appelées propositions, de constantes, de connecteurs et de parenthèses. Les constantes prenant les valeurs de vérités vraies ou fausses, soient respectivement 1 ou 0. Les principaux connecteurs sont la disjonction $\vee$, la conjonction $\wedge$ et la négation $\neg$. S'ajoutent les constantes tautologiques (constantes toujours vraies) $\top$ et de contradiction $\perp$. La relation d'implication s'exprime par $\rightarrow$ et celle d'équivalence par $\leftrightarrow$.

Le langage de la logique propositionnelle comporte également des formules bien formées.

L'ensemble des formules bien formées constitue le plus petit ensemble tel que :

Définition 2.1. – *Les constantes tautologiques $\top(1)$ ou $\bot(0)$ sont des formules.*

– *Une variable propositionnelle telle que P ou Q constitue une formule.*

– *Si P ou Q sont des formules alors les expressions logiques $P \vee Q$, $\neg P \wedge Q$, $P \rightarrow Q$, $\neg P \leftrightarrow Q$ sont également des formules.*

Par exemple, soit $P =$ «Il y a une route» et $Q =$ «Il y a une autoroute». Avec ces égalités, l'expression $P \vee Q$ se lit : «Il y a une route ou il y a une autoroute.».

2. Aspect axiomatique de la logique propositionnelle

L'aspect axiomatique consiste à définir un système formel à travers lequel les déductions effectuées mènent à établir des théorèmes. En logique, la définition d'un système formel impose l'utilisation d'axiomes qui sont des propositions évidentes ne nécessitant pas de démonstration et qui sont considérées comme acquises. Plusieurs auteurs ont proposé des systèmes d'axiomes dont Whitehead et Russell [WR12] et Hilbert et Ackermann [HA28]. Pour nos besoins et des raisons de simplicité, nous retenons les axiomes de Lukasiewicz, remontant à l'année 1930, qui sont les suivants :

1. $P \rightarrow (Q \rightarrow P)$
2. $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$
3. $(\neg P \rightarrow \neg Q) \rightarrow (Q \rightarrow P)$

Ces axiomes se situent à la base de la programmation logique et servent à résoudre des programmes logiques. Le système formel de Lukasiewicz comporte uniquement deux règles d'inférence, soient la substitution et le modus ponens.

Le symbole d'assertion logique "$\vdash$" dans l'expression "$\vdash P$" signifie qu'il existe une preuve de la formule P qui constitue un théorème. L'expression $A, B, C \vdash P$ se lit : "Les axiomes A, B, C constituent des axiomes desquels découle la conséquence logique que P est un théorème". Le symbole $\rightarrow$ signifie "implique".

Ex : $\vdash (P \rightarrow (Q \rightarrow P))$

Définition 2.2. *Une théorie est un ensemble fini ou infini de formules propositionnelles.*

Définition 2.3. *Dans le cadre d'une théorie, un théorème est une formule que l'on peut démontrer comme étant vraie, en partant des axiomes et en procédant par déduction.*

Définition 2.4. *La preuve ou la démonstration d'une formule est la liste des étapes, non ambiguës (en utilisant les règles d'inférences si on travaille dans un système formel), qui permettent de passer d'une formule supposée vraie à une autre. On obtient alors la preuve que la formule obtenue est vraie.*

Définition 2.5. *Dans le cadre d'un système formel, les règles d'inférences ou les règles de transformation sont des règles précises, applicables mécaniquement, qui permettent de transformer une formule en une autre, c'est à dire de déduire d'une formule une autre formule.*

Définition 2.6. *Nous affirmons qu'une théorie formelle est consistante si elle n'admet aucune formule F telle que F et son contraire $(\neg F)$ puissent être démontrés.*

La règle de dérivation ou du modus ponens

La règle du modus ponens permet de tirer des conclusions à partir d'un ensemble de formules. Le *modus ponens* est une forme de raisonnement logique qui consiste à affirmer un jugement hypothétique. (« si A alors B ») et à poser ensuite la prémisse (« or, A ») pour poser comme conclusion la conséquence du jugement hypothétique (« donc B »). Le modus ponens se lit comme suit : «Si A implique B, or nous avons A, donc nous avons B.».

De manière formelle, il s'exprime comme suit : $\frac{\vdash A,\ \vdash A \to B}{\vdash B}$

Par exemple, si A=«Il y a une autoroute» et B=«Il y a une route connectée à cette autoroute». Le modus ponens se lit tel qu'il suit : «Si il y a une autoroute, cela implique qu'il y a une route connectée à cette autoroute, or nous avons une autoroute, donc nous avons qu'il y a une route connectée à cette autoroute».

La règle de substitution

La règle de substitution autorise à remplacer à l'intérieur d'un théorème toutes les instances d'une variable propositionnelle par une autre variable propositionnelle ou encore une formule bien formée.

Par exemple, Soit le théorème :

$\vdash ((P \rightarrow Q) \rightarrow ((Q \rightarrow R) \rightarrow (P \rightarrow R)))$

Nous appliquons la règle de substitution en substituant les valeurs P pour A, Q pour B et R pour C.

Nous obtenons alors le théorème :

$\vdash ((A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C)))$

Voici quelques théorèmes utiles pour la déduction de conclusions :

$\vdash ((P \rightarrow Q) \rightarrow ((Q \rightarrow R) \rightarrow (P \rightarrow R)))\,;$

$\vdash (P \rightarrow ((P \rightarrow Q) \rightarrow Q))\,;$

$\vdash (\neg P \rightarrow (P \rightarrow Q))\,;$

$\vdash (\neg\neg P \rightarrow P)\,;$

$\vdash (P \rightarrow \neg\neg P)\,;$

$\vdash ((P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P))\,;$

$\vdash (P \rightarrow (\neg Q \rightarrow \neg(P \rightarrow Q)))\,;$

$\vdash ((Q \rightarrow P) \rightarrow ((\neg Q \rightarrow P) \rightarrow P))\,;$

3. Aspect sémantique de la logique propositionnelle

L'aspect sémantique de la logique propositionnelle vise l'interprétation des formules de $\mathcal{L}$ et consiste à analyser des formules qui sont toujours vraies, nommées tautologies. En d'autres mots, elle vise l'étude des propositions et des ensembles de propositions. Ces derniers sont définis à l'aide d'interprétations. Ainsi, afin de donner un sens à des énoncés logiques, il faut étudier les conditions de vérité de ces énoncés. Une interprétation est une fonction associant à chaque variable propositionnelle d'un langage propositionnel fini une valeur de vérité, soit vraie soit fausse.

La définition qui suit présente de façon plus générale ce qu'est une interprétation.

Définition 2.7. *Une interprétation σ constitue une assignation de valeurs de vérités aux différentes variables propositionnelles d'une formule ou d'un ensemble de formules.*

De manière plus formelle, une interprétation est une application σ de P dans 0,1 telle que $\sigma(0) = 0$ et $\sigma(1) = 1$. L'application σ est étendue aux formules de la manière suivante : $\forall P, Q \in \mathcal{L}$

- $\sigma(\neg P) = 1$ - $\sigma(P)$;
- $\sigma(P \vee Q) = max(\sigma(P), \sigma(Q))$;
- $\sigma(P \wedge Q) = min(\sigma(P), \sigma(Q))$;
- $\sigma(P \rightarrow Q) = max((1 - \sigma(P)), \sigma(Q))$;
- $\sigma(P \leftrightarrow Q) = min(max((1 - \sigma(P)), \sigma(Q)), max(\sigma(P), (1 - \sigma(Q))))$.

Par exemple, soit la formule $\varphi = (a \vee b) \wedge c$, un exemple d'interprétation associé à cette formule est :$\sigma(a) = 1$, $\sigma(b) = 0$, $\sigma(c) = 1$. Ainsi, $\sigma(\varphi) = 1$.

Les résultats des interprétations des formules sont souvent compilés à l'intérieur de tables de vérité telles que la suivante :

Voici quelques faits généraux importants caractérisant la sémantique de la logique propositionnelle :

1. L'interprétation d'une proposition associe à toute formule propositionnelle une et une seule valeur de vérité.
2. L'interprétation correspond à une possibilité logique ainsi qu'à une ligne dans une table de vérité.
3. Une formule propositionnelle constitue une tautologie si et seulement si elle est vraie pour toutes les interprétations possibles de ses éléments les plus simples.

A	$\rightarrow$	B
1	1	1
1	0	0
0	1	1
0	1	0

FIG. 2.1 – Table de vérité de l'implication

4. Une formule constitue une contradiction dans le cas où aucune interprétation ne parvient à rendre «vraie» cette première.
5. Un ensemble de propositions est satisfaisable si et seulement si il existe une interprétation qui rend valide toutes les propositions de ce même ensemble.
6. Une formule propositionnelle est une conséquence sémantique d'un ensemble de propositions si et seulement si toute interprétation validant la totalité des formules propositionnelles d'un tel ensemble valide aussi cette formule.

Définition 2.8. *Une tautologie est toute formule P, telle que pour toute interprétation σ, $\sigma(P) = 1$. Une tautologie se note $\models P$.*

Définition 2.9. *Une formule Q est une conséquences logique de la formule P si $\sigma(P) = 1$ alors $\sigma(Q) = 1$, il suit que $P \models Q$.*

Définition 2.10. *Une formule Q constitue une conséquence logique d'un ensemble de formules A, si $\forall P \in A$, $\sigma(P) = 1$ alors $\sigma(Q) = 1$, il suit que $A \models Q$.*

Définition 2.11. *Deux formules P et Q sont considérées équivalentes si $P \models Q$, $Q \models P$, il suit que $P \equiv Q$.*

Définition 2.12. *Une formule P est considérée cohérente s'il existe une interprétation σ tell que $\sigma(P) = 1$. Dans ce cas, σ devient un modèle de P et il découle que $\sigma \models P$.*

Définition 2.13. *Un ensemble de formules A est considéré vérifiable ou cohérent si il existe une interprétation σ qui est telle que $\forall P \in A$, $\sigma(P) = 1$. Il suit que σ est un modèle de A, ce qui se note $\sigma \models A$.*

Définition 2.14. *Deux formules sont équivalentes si elles possèdent exactement les mêmes modèles.*

Les théorèmes qui suivent sont liés à la sémantique de la logique propositionnelle.

Le théorème d'adéquation exprime que les théorèmes sont toujours vrais et il s'énonce de la manière suivante :

THÉORÈME D'ADÉQUATION. $\forall P \in \mathcal{L}$, *si* $\vdash P$ *alors* $\models P$. *Tous les théorèmes constituent des tautologies.*

Le théorème de complétude exprime que les énoncés toujours vrais sont des théorèmes :

THÉORÈME DE COMPLÉTUDE. $\forall P \in \mathcal{L}$, *si* $\models P$ *alors* $\vdash P$. *Toutes les tautologies constituent des théorèmes.*

D'un théorème, il n'est pas possible de conclure une formule et la négation de cette même formule simultanément, ce qu'énonce de théorème de cohérence :

THÉORÈME DE COHÉRENCE. *Il est impossible d'avoir simultanément* $\vdash P$ *et* $\vdash \neg P$.

Le théorème de complétude est aussi valide pour les ensembles de formules. Dans ce cas, il s'énonce comme suit :

THÉORÈME DE COMPLÉTUDE GÉNÉRALISÉE. *Soit* A *un ensemble de formules de* $\mathcal{L}$, *soit* P *une formule de* $\mathcal{L}$, $A \models P$ *tel que* $A \vdash P$.

Le théorème de compacité exprime qu'il est possible de donner des valeurs de vérité aux formules constituant la proposition de façon à ce que celle-ci prenne la valeur vraie. Il s'énonce de la manière suivante :

THÉORÈME DE COMPACITÉ. *Soit* A *un ensemble de formules de* $\mathcal{L}$, *si pour toute famille finie* A', $A' \subset A$, *il existe une interprétation* σ *telle que* $\forall P \in A'$, $\sigma(P) = 1$ *alors il existe une interprétation telle que* $\forall P \in A$, $\sigma(P) = 1$.

Le théorème de finitude montre que si une formule P peut être déduite, alors la formule P peut être déduite à partir d'un ensemble fini de formules.

THÉORÈME DE FINITUDE. *Soit* A *un ensemble de formules de* $\mathcal{L}$, *si* $A \models P$, *alors il existe* A' *fini tel que* $A' \models P$.

Le théorème de décidabilité nous informe qu'il est possible de déduire si oui ou non une formule est vraie si cette formule appartient à un ensemble fini de formules. Ce théorème s'énonce comme suit :

THÉORÈME DE DÉCIDABILITÉ. *$\forall P \in \mathcal{L}$, il existe un programme qui pour toute formule P, informe en un temps fini si oui ou non $\vdash P$.*

Définition 2.15. *Un littéral constitue une proposition ou la négation d'une proposition.*

Par exemple, soit a une proposition. Dans ce cas, a est donc un littéral positif. Dans le cas où nous aurons une expression de la forme $\neg a$, nous affirmerons qu'il s'agit d'un littéral négatif.

Définition 2.16. *La disjonction logique en logique propositionnelle et éventuellement en logique des prédicats constitue un "ou" inclusif. Ainsi, dans l'expression, $a \vee b$, "$\vee$" est inclusif, ce qui revient à dire que pour valider cette expression, a peut être vrai, b peut être vrai, ou encore a et b peuvent être vrais.*

Définition 2.17. *Si une interprétation rend une formule valide, nous dirons que l'interprétation m est un modèle de f, qui est notée : $m \models f$.*

Définition 2.18. *Un contre-modèle constitue une interprétation ne validant pas une formule.*

Ex :$\varphi = \varphi_1 \wedge \varphi_2 = (a \vee b) \wedge c$ où $\varphi_1 = (a \vee b)$ et $\varphi_2 = c$, Interprétation : $I(a) = 1, I(b) = 0$ *et* $I(c) = 0$. Ainsi, $I(\varphi) = 0$. Donc la formule n'est pas validée.

Définition 2.19. *Un modèle partiel est un modèle validant une formule à l'intérieur d'un ensemble de formules. Ex : $m \models \varphi_1$ (par rapport à φ_1).*

Définition 2.20. *Une clause est formée d'une disjonction de littéraux qui sont des propositions.*

Définition 2.21. *Un atome, en logique propositionnelle, est une proposition qui ne contient pas de connecteur logique.*

Définition 2.22. *Une formule est un énoncé logique composé d'un ou plusieurs atomes.*

Par exemple, a, b et d sont des atomes.

Définition 2.23. *Une formule est exprimée sous la forme conjonctive normale si celle-ci est exprimée sous forme de conjonction de clauses.*

Voici un exemple de formule exprimée sous la forme conjonctive normale : $\varphi = (a \vee b) \wedge (c \vee d) \wedge (e \vee f)$. Le principal intérêt de la forme conjonctive normale est que sous cette forme, des ensembles de clauses se prêtent bien à la résolution.

Une clause peut aussi être exprimée de manière équivalente sous forme de règle en utilisant les connecteurs $\neg$, $\vee$ et $\wedge$ comme dans l'exemple suivant : la règle $c \leftarrow a \wedge b$ est équivalente à la clause $c \leftarrow \neg a \wedge \neg b$.

Une clause qui contient au plus un littéral positif est une *clause de Horn*. Par exemple, $\neg a \vee \neg b \vee \neg c \vee d$ est une clause de Horn puisqu'elle ne contient qu'un seul littéral positif, soit d. En opposition, $\neg a \vee \neg b \vee c \vee d$ n'est pas une clause de Horn puisqu'elle contient deux littéraux positifs qui sont c et d. La clause triviale se note $\neg b \vee b$ puisqu'elle peut être vraie ou fausse et la clause vide notée $\sqcap$ est engendrée par $\neg b \wedge b$ laquelle expression est vraie et fausse simultanément, ce qui constitue une contradiction.

Voici un exemple de formule bien formée à partir d'un ensemble de formules élémentaires pouvant être exprimées dans le cadre délimité par le langage $\mathcal{L}$:

φ : x est une route

ψ : x est une autoroute

μ : x appartient à un réseau routier

$\varphi \vee \psi \rightarrow \mu$

Afin de dresser un portrait des principales propriétés qui caractérisent le calcul propositionnel, nous présentons les principales tautologies propres à ce calcul. Une tautologie est une formule qui est toujours vraie pour une assignation de valeurs dans une formule, une telle assignation de valeur à l'intérieur d'une formule F se note $\models$ F. Les premières tautologies présentées par Louis Gacôgne [Gac97] qui se situent à la base du calcul propositionnel sont les suivantes :

- Involutivité de la négation $\neg(\neg P) \leftrightarrow P$.
- Les lois de Morgan, $\neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$ et $\neg(P \vee Q) \leftrightarrow \neg P \wedge \neg Q$.
- La commutativité des connecteurs $\wedge$ et $\vee$: $P \wedge Q \leftrightarrow Q \wedge P$ et $P \vee Q. \leftrightarrow Q \vee P$.
- L'idempotence : $P \wedge P \leftrightarrow P$ et $P \vee P \leftrightarrow P$.
- L'associativité : $P \wedge (Q \wedge R) \leftrightarrow (P \wedge Q) \wedge R$ et $P \vee (Q \vee R) \leftrightarrow (P \vee Q) \vee R$.

– Les distributivités mutuelles : $P \wedge (Q \vee R) \leftrightarrow (P \wedge Q) \vee (P \wedge R)$ et $P \vee (Q \wedge R) \leftrightarrow (P \vee Q) \wedge (P \vee R)$.

– Éléments neutre et absorbant : Soit 0 et 1 désignant respectivement le faux et le vrai, P étant une proposition quelconque :

– Dans les propriétés $0 \wedge P \leftrightarrow 0$ et $0 \vee P \leftrightarrow P$, 0 est l'élément absorbant pour $\wedge$, et l'élément neutre pour $\vee$.

– Dans les propriétés $1 \wedge P \leftrightarrow P$ et $1 \vee P \leftrightarrow 1$, 1 est l'élément neutre pour $\wedge$, et l'élément absorbant pour $\vee$.

– La complémentation : $P \wedge \neg P \leftrightarrow 0$ et $P \vee \neg P \leftrightarrow 1$.

– La contraposition : $(\neg Q \rightarrow \neg P) \leftrightarrow (P \rightarrow Q)$.

Ces tautologies sont très utiles afin de résoudre des problèmes de satisfaction de contrainte (CSP) en logique .

2.2.2 La logique des prédicats

La logique des prédicats est munie d'une syntaxe, d'un langage, d'une axiomatique et d'une sémantique. Dans ce qui suit, nous présentons ces différentes aspects de la logique des prédicats. Après la section sur le langage de la logique des prédicats, nous résumons les principaux points qui distinguent les langages de la logique propositionnelle et de la logique des prédicats.

1. Aspect syntaxique des prédicats

Le langage de la logique des prédicats

Le langage de la logique des prédicats comporte plusieurs ressemblances avec celui de la logique propositionnelle. Ce langage est constitué d'un ensemble infini dénombrable de symboles prédicatifs aussi appelés prédicats, d'un ensemble infini dénombrable de symboles fonctionnels, d'un ensemble infini dénombrable de variables, de connecteurs et de quantificateurs. Les connecteurs sont les mêmes que pour la logique propositionnelle soient la disjonction $\vee$, la conjonction $\wedge$, la négation $\neg$, l'implication $\rightarrow$ et la bi-implication aussi appelée équivalence $\leftrightarrow$. Il en est de même pour les constantes tautologiques $\top$ et de contradiction $\perp$ et pour les parenthèses. Les quantificateurs existentiel $\exists$ et universel $\forall$ viennent s'ajouter. Dans les lignes qui suivent, des définitions importantes liées au langage de la logique des prédicats sont présentés :

Définition 2.24. *Un symbole fonctionnel représente une fonction constituant un terme.*

Définition 2.25. *Un prédicat est une expression logique dont la valeur peut être vraie ou fausse selon la valeur des arguments.*

Par exemple dans $P(t_1)$, P est un prédicat dont la valeur dépend du terme en argument t_1.

Définition 2.26. *Par terme, nous entendons :*

- *Une variable x est un terme ;*
- *Un symbole fonctionnel f est un terme ;*
- *Si $t_1, ..., t_n$ sont des termes, $f(t_1, ..., t_n)$ est un terme.*

Par exemple a et $A(t)$ sont des termes.

Définition 2.27. *Un atome est une expression de la forme $P(t_1, ..., t_n)$ où $t_1, ..., t_n$ sont des termes et P est un prédicat. Ex :$A(t_1)$ est un atome car $A(t_1)$ a la même forme que $P(t_1)$.*

Définition 2.28. *Un symbole fonctionnel constitue un symbole représentant une fonction.*

Définition 2.29. *Par clause, nous entendons une disjonction de littéraux formée d'atomes. Ex : $A(t_1) \vee \neg A(t_1) \vee \neg B(t_2)$.*

Définition 2.30. *Une formule en logique des prédicats est un énoncé logique composé d'un ou plusieurs atomes. Ex :$A(t_1)$.*

De manière plus formelle, nous entendons par formule :

- un atome est une formule ;
- si A et B sont des formules alors $\neg A$, $A \vee B$, $A \wedge B$, $A \leftrightarrow B$ sont des formules ;
- Si A est une formule et x une variable alors $\forall xA$, $\exists xA$ sont des formules.

Note : Une formule bien formée respecte les contraintes imposées par la syntaxe du langage contrairement à une formule malformée. Par formule close ou fermée, nous entendons une formule qui ne comporte pas de variables libres, soit de variables qui ne sont pas sous la portée de quantificateurs.

Formule : Énoncé logique composé d'un ou plusieurs atomes. Ex : *A(t1)* .

Définition 2.31. *L'unification consiste à remplacer une variable dans tous les arguments d'une expression par une constante. Ex : Soit la formule* $\forall T, A(T) \vee B(T)$*, elle devient par unification de a, l'expression* $\forall a, A(a) \vee B(a)$*.*

Comparaison entre le langage de la logique propositionnelle et le langage de la logique des prédicats

Les langages de la logique propositionnelle et de la logique des prédicats se réjoignent et diffèrent à plusieurs niveaux. Voici une liste des principales distinctions qui existent entre le langage de la logique propositionnelle et celui de la logique des prédicats :

– La logique propositionnelle ne possède pas de symboles fonctionnels (termes) tels que $A(t)$ alors que la logique des prédicats en possède.
– En logique propositionnelle, il n'est possible que de quantifier sur les variables alors qu'en logique des prédicats, il est possible de quantifier sur les variables et sur les prédicats.

Le nombre d'arguments dans un prédicat s'appelle l'arité.

Par exemple, un prédicat $P(t_1, t_2)$ est un prédicat d'arité 2.

Une proposition P est en fait un prédicat d'arité 0.

Soit une signature $\sum$ contenant :

– Un ensemble dénombrable de fonctions $\sum F = \{f, g, h, ...\}$ chacun possédant une arité déterminée.
– Un ensemble dénombrable de prédicats $\sum P = \{p, q, r, ...\}$ chacun possédant une arité déterminée.

Le calcul propositionnel peut être vu sous forme d'un calcul des prédicats sur une signature $\sum$ si :

1. l'ensemble $\sum F$ est vide.
2. l'ensemble $\sum P$ contient uniquement des prédicats d'arité 0.
3. les quantificateurs ne sont pas utilisés.

2. Aspect axiomatique des prédicats

L'aspect axiomatique de la logique des prédicats se résume essentiellement à la définition d'un système formel de déduction à travers lequel il est possible d'effectuer des déductions dont les résultats sont des théorèmes.

Ces axiomes se situent à la base de la programmation logique et servent à résoudre des programmes logiques. Voici les axiomes de l'extension à la logique des prédicats du système de Lukasiewisz pour la logique des prédicats que nous retenons pour des questions de simplicité :

$(P \rightarrow (Q \rightarrow P))$;

$(P \rightarrow (Q \rightarrow R)) \rightarrow (P \rightarrow Q) \rightarrow (P \rightarrow R)$;

$((\neg P \rightarrow \neg Q) \rightarrow (Q \rightarrow P))$;

$(\forall x P(x) \rightarrow P(t))$;

$((S \rightarrow Q) \rightarrow (S \rightarrow \forall x Q))$.

Le modus ponens se définit de la même manière qu'en logique propositionnelle, soit tel qu'il suit : $\frac{\vdash P,\, \vdash P \,\rightarrow\, Q}{\vdash Q}$

La règle de généralisation s'exprime de la façon suivante :

$\frac{\vdash P}{\vdash \forall x P}$

La règle de substitution

La règle de substitution autorise à remplacer à l'intérieur d'un théorème toutes les instances d'une variable propositionnelle par une autre variable propositionnelle ou encore une formule bien formée.

3. Aspect sémantique des prédicats

L'aspect sémantique de la logique des prédicats se résume à l'interprétation des formules de $\mathcal{L}_{\mathcal{P}}$.

Une interprétation en logique des prédicats constitue un triplet $I = (D, I_c, I_v)$ dans lequel D est un ensemble non vide correspondant au domaine d'interprétation. Dans celui-ci, I_c est une fonction qui associe à tout symbole fonctionnel une valeur du domaine D et à tout prédicat une valeur de 0 ou 1. I_v est une fonction associant à toute variable une valeur propre au domaine D.

Interprétation : Une interprétation constitue une fonction associant à chaque variable propositionnelle de $I(A)$ à A comme suit :

- Si x est une variable libre alors $I(x) = I_v(x)$;
- $I(f(t_1, ..., t_n))= (I_c(f))(I(t_1), ..., I(t_n)$;
- $I(P(t_1, ..., t_n))= (I_c(P))(I(t_1), ..., I(t_n)$;
- Si A et B sont des formules alors $\neg A$, $A \vee B$, $A \wedge B$, $A \leftrightarrow B$ doivent être interprétés comme des formules de logique propositionnelle ;
- Si A est une formule et x une variable alors $I(\forall x) = 1$ si $I_{x/d}(A) = 1$ pour tout élément $d \in D$;
- Si A est une formule et x une variable alors $I(\exists x) = 1$ si $I_{x/d}(A) = 1$ pour tout élément $d \in D$.

Les théorèmes qui suivent sont liés à la sémantique de la logique des prédicats.

Le théorème d'adéquation exprime que les théorèmes sont toujours vrais et il s'énonce de la manière suivante :

THÉORÈME D'ADÉQUATION. *$\forall P \in \mathcal{L_P}$, si $\vdash P$ alors $\models P$. Tous les théorèmes constituent des tautologies.*

Le théorème de complétude exprime que les énoncés toujours vrais sont des théorèmes :

THÉORÈME DE COMPLÉTUDE. *$\forall P \in \mathcal{L_P}$, si $\models P$ alors $\vdash P$. Toutes les tautologies constituent des théorèmes.*

D'un théorème, il n'est pas possible de conclure une formule et la négation de cette même formule simultanément, ce qu'énonce de théorème de cohérence :

THÉORÈME DE COHÉRENCE. *Il est impossible d'avoir simultanément $\vdash P$ et $\vdash \neg P$.*

Le théorème de complétude est aussi valide pour les ensembles de formules. Dans ce cas, il s'énonce comme suit :

THÉORÈME DE COMPLÉTUDE GÉNÉRALISÉE. *Soit A un ensemble de formules de $\mathcal{L_P}$, soit P une formule de $\mathcal{L_P}$, $A \models P$ tel que $A \vdash P$.*

Le théorème de compacité exprime qu' il est possible de donner des valeurs de vérité aux formules constituant la proposition de façon à ce que celle-ci prenne la valeur vraie. Il s'énonce de la manière suivante :

THÉORÈME DE COMPACITÉ. *Soit A un ensemble de formules de $\mathcal{L}$, si pour toute famille finie A', $A' \subset A$, il existe une interprétation σ telle que $\forall P \in A'$, $\sigma(P) = 1$ alors il existe une interprétation telle que $\forall P \in A$, $\sigma(P) = 1$.*

Le théorème de finitude montre que si une formule P peut être déduite, alors la formule P peut être déduite à partir d'un ensemble fini de formules.

THÉORÈME DE FINITUDE. *Soit A un ensemble de formules de $\mathcal{L}$, si $A \models P$, alors il existe A' fini tel que $A' \models P$.*

Le théorème de décidabilité nous informe qu'il est possible de déduire si oui ou non une formule est vraie si cette formule appartient à un ensemble fini de formules. Ce théorème s'énonce comme suit :

THÉORÈME DE DÉCIDABILITÉ. *$\forall P \in \mathcal{L}$, il existe un programme qui pour toute formule P, informe en un temps fini si oui ou non $\vdash P$.*

Une théorie formelle est complète si il n'existe pas de formule F telle que ni F ni son contraire $(\neg F)$ ne puissent être démontrées, c'est à dire si tout ce qui est vrai dans la théorie est démontrable à l'intérieur de la théorie.

2.2.3 La logique possibiliste

La logique possibiliste est une logique qui permet d'associer un degré de possibilités à des interprétations. Cette logique présente un intérêt particulier pour l'établissement d'ordres de préférences en logique. Elle sera très utile, lors de situations où des règles devront avoir prévalence sur d'autres.

Soit π une distribution de possibilité. Nous appelons $\pi(\omega)$ le degré de possibilité lié à chaque interprétation w. Nous définissons un ensemble d'interprétations Ω.

Le degré de possibilité associé à la distribution de possibilités π dans ω se note Π_π.

Nous définissons le degré de nécessité par N_π . Le degré de nécessité indique à quel point φ peut être inféré par π.

L'inférence sémantique issue de π s'écrit $\models_\pi$. K_{pos} représente une base de croyances possibiliste qui est un ensemble de formules $K_{pos} = \{(\varphi_i, \alpha_i) : i = 1, ..., n\}$, où φ_i est une formule propositionnelle et $\alpha_i \in [0, 1]$.

$\alpha - coupe$ se définit comme l'ensemble des formules pondérées de K_{pos} dont le poids est supérieur à α. Il se note : $K_{pos,\geq\alpha}$.

Le degré d'incohérence associé à K_{pos} se note $Inc(K_{pos})$.

2.2.4 Déduction automatique

Avant de présenter la logique du premier et du seconde ordre, nous présentons, dans ce qui suit, deux définitions de notions importantes, se situant à la base du mécanisme du raisonnement déductif en logique du premier et du second ordre. Ces deux définitions en faciliteront sa compréhension par la suite.

Définition 2.32. *Une théorie est un ensemble fini ou infini de formules propositionnelles.*

Définition 2.33. *Dans une théorie, un axiome est une formule de base, que l'on considère vraie sans démonstration. C'est en quelque sorte le "point de départ" qui servira à démontrer des théorèmes.*

2.2.5 Logique du premier et du second ordre

La *logique du premier* ordre également appelée *calcul des prédicats du premier ordre* constitue une formalisation du langage des mathématiques introduite par les logiciens du début du 20e siècle. En logique du premier ordre, il est possible de quantifier sur les variables mais il est impossible de quantifier sur les prédicats.

Dans l'exemple ci-dessous, nous voyons une quantification sur une variable qui porte sur une porte :

exemple : Soit p une porte correspondant à une instance d'une variable X.

Soit g un garage correspondant à une instance d'une variable X.

Soit M un prédicat représentant une maison.

Soit l'atome $M(X)$ lequel se lit «Il existe X sur la maison».

$\exists p\ M(p)$ se lit "Il existe une porte sur la maison"

Pour pouvoir quantifier sur des prédicats en plus de sur des variables, il est utile de recourir à la logique du deuxième ordre.

Ainsi, dans l'exemple qui suit, nous voyons qu'en logique du deuxième ordre il est également possible de quantifier un prédicat en plus de pouvoir quantifier sur des variables :

exemple : $\exists M, M(p)$.
$\exists M(M(p) \wedge M(g))$. se lit «Il existe une porte sur la maison et un garage dans la maison.»

note : La formule logique $\exists M, M(p)$. n'est pas permise en logique du premier ordre mais elle l'est en logique du deuxième ordre.

2.2.6 Synthèse sur les logiques existantes et le raisonnement logique dans un contexte de géomatique

Il existe un grand nombre de logiques. Parmi les principales, il existe la logique classique, la logique des défauts, la logique possibiliste et la logique floue. Les logiques peuvent être monotones ou non monotones. Elles possèdent toutes des forces et des lacunes. Elles ont un point en commun qui est qu'elles sont toutes difficiles à implanter compte tenu de leur étendue et leur complexité. De façon générale, dans un contexte de géomatique, ce sont les implantations des logiques qui nous intéressent le plus, puisque ces implantations peuvent servir dans des contextes d'applications propres à la géomatique. Certains mécanismes de raisonnement s'apparentant à ceux présents dans les logiques peuvent être recréés comme c'est le cas pour la logique des défauts qui est liée au raisonnement par défaut. Ainsi, une logique est avantageuse, par rapport à une autre dans un contexte d'application, lorsqu'elle permet de recréer le type de raisonnement dont on a besoin. Il découle dans le contexte de cette recherche, puisque nous nous intéressons au traitement de cas d'exceptions, que la logique des défauts est celle qui présente le plus grand intérêt pour nous puisqu'elle est liée au raisonnement par défaut qui peut être recréé en programmation logique et que ce dernier type de raisonnement permet de traiter de l'information évolutive. Le raisonnement en logique classique ne permet pas de réviser des conclusions tirées antérieurement, ce qui devient très limitatif en géomatique où l'information évolue constamment et où les conclusions de raisonnements sont susceptibles de varier régulièrement. La logique possibiliste présente un intérêt particulier lorsque l'on souhaite traiter des informations qui ont des niveaux d'importance différents et donc des poids différents. Ce type de logique est très intéressant pour la géomatique et il en existe plusieurs implantations. En effet, il est fréquent que des informations n'aient pas le même niveau d'importance en géomatique. La logique floue est intéressante pour certaines applications en photogrammétrie et en traitement d'image puisqu'elle permet de traiter des informations auxqu'elles des degrés de vérité fractionnaires sont associés. Or, des associations peuvent souvent être établies entre des valeurs de pixels et des degrés de valeurs de vérités fractionnaires. Ainsi, chaque logique possède l'avantage de pouvoir modéliser un type spécifique de raisonnement.

2.2.7 Programmation logique et modèles stables

La programmation logique se situe à la base du fonctionnement du *PROLOG* et de l'*ASP*. Le concept de modèle stable intervient en *ASP* et sert, dans le contexte de ce mémoire, à détecter des ensembles d'informations incohérentes entre elles.

Définition 2.34. *Un programme logique P constitue un ensemble de règles r de la forme :*

$$l_0 \leftarrow l_1, l_2, ..., l_m,\ not\ l_n, l_{n+1}, l_{n+2}, ..., l_m$$

Dans ce programme, l_0 constitue la tête de r. $corps^{+}(r) = \{l_1, l_2, ..., l_m\}$ correspond à l'ensemble des prérequis de r alors que $corps^{-}(r) = l_n, l_{n+1}, l_{n+2}, ..., l_m$ correspond à l'ensemble des bloqueurs de r.

Définition 2.35. *Soit Δ un programme logique et X un sous-ensemble de littéraux $Litt\Delta$. On affirmera que X est un modèle stable de Δ si et seulement si X est égale à la clôture déduite de Δ^X. i.e. $X = Cn(\Delta^X)$.*

Ex : Soit un programme Δ composé d'un fait et de deux règles :

$p.$
$q :- p\ ,\ not\ r.$
$r :- p\ ,\ not\ q.$

note : L'implication de note : −.

Deux modèles stables de Δ sont : $\{q, p\}$ et $\{r, p\}$.

2.2.8 Langages de programmation logique

Le langage de programmation PROLOG

Le *PROLOG* est un langage de programmation reposant sur les clauses de Horn. Toute formule de logique classique du premier ordre peut être ramenée sous forme de clause de Horn.

Une clause prendra le nom de fait, de règle ou de requête suivant les valeurs d'indices de m et n dans l'expression clausale ci-dessous :

$\neg A_1 \vee \neg A_2 ... \vee \neg A_m \vee B_n$

Ainsi,

1. si $m = 0$ et $n = 1$, il s'agit d'un fait. B_1
2. si $m > 0$ et $n = 1$, il s'agit d'une règle. $\neg A_1 \vee ... \neg A_m \vee B_1$
3. si $m > 0$ et $n = 0$, il s'agit d'un but. $\neg A_1 \vee ... \neg A_m$

Le *PROLOG* permet d'effectuer des requêtes en donnant un but à la machine *PROLOG* et celle-ci détermine en procédant par chaînage arrière si ce but est une conséquence du programme logique. Donner un but à la machine *PROLOG* revient à effectuer une requête.

Voici les principales étapes impliquées lors d'une requête simple effectuée en *PROLOG* :

Le moteur de raisonnement *PROLOG* tente de trouver une preuve que la requête est une relation qui existe à l'aide de la procédure suivante :

prouver(requête) :

1) Pour chaque fait et tête de règle qui concorde avec la requête :

1. Si un fait : on a trouvé une preuve
2. Si une tête de règle :
   Pour chaque condition de cette règle : prouver(condition).

2) Si les conditions précédentes sont vérifiées, alors nous avons trouvé une preuve. Sinon, nous n'avons pas trouvé de preuve.

Exemple d'exécution : A. : 1) A est un fait, on a trouvé une preuve qui est «A».
B. : 1) B est un fait, on a trouvé une preuve qui est «B».
$C \leftarrow A, B.$: 1) C n'est pas un fait.
1)1. C est une tête de règle. A est une preuve d'une condition de la règle $C \leftarrow A, B.$ et B également. Les deux conditions de la règle $C \leftarrow A, B.$ sont donc vérifiées.
2) Nous avons donc une preuve de C.

Les faits et les règles sont examinés dans l'ordre d'apparition dans le programme. Si une sous-preuve échoue, la recherche revient sur ses pas (retour-arrière ou "backtracking") pour tenter la prochaine alternative de preuve.

Backtracking

Le backtracking classique aussi appelé bactracking chronologique s'effectue en instanciant chronologiquement de manière sucessive les variables du problème, et ce, en suivant un ordre prédéfini. Cette technique vise à trouver des modèles pour des interprétations données.

Langage de programmation logique ASP

L'*ASP* est un formalisme de représentation logique avec sémantique de modèles stables (*Answer Set Programming*) qui rend possible la mise en oeuvre de raisonnement logique non monotone. Nous nous servirons de ce formalisme afin de représenter des géocroyances propres à la BNDT. Il s'apparente considérablement au *PROLOG* dans son écriture. L'écriture de règles et de faits élémentaires a la même forme qu'en *PROLOG*. S'il comporte l'avantage, lorsqu'il est employé de pair avec *Smodels*, de permettre d'inférer des disjonctions de littéraux, par contre, l'ASP n'inclut pas d'objets comme des "listes" et d'autres objets très utiles qu'offrent le *PROLOG*. Il n'est pas dit cependant que la conception d'extensions à *Smodels* ne pourrait pas être envisagée pour combler de tels besoins.

De façon générale, l'écriture de l'*ASP* prend la forme suivante :

Soient A et B deux littéraux, et m et n deux indices.

1. si $m = 0$ et $n = 1$, il s'agit d'un fait. B_1
2. si $m > 0$ et $n \geq 1$, il s'agit d'une règle. $\neg A_1 \vee ... \vee \neg A_m \vee B_1 \vee ... \vee B_n$

Le problème SAT

Le problème de satisfaisabilité (SAT) est un problème de décision en théorie calculatoire qui consiste à vérifier si une formule donnée de la logique propositionnelle exprimée sous la forme normale conjonctive est satisfaisable. Cook a démontré que le problème SAT est non-déterministe polynomial complet (NP-complet), laquelle preuve est présentée, entre autres, dans [GJ79]. La résolution du problème SAT est particulièrement intéressante puisque ce problème est lié sur le plan logique au fonctionnement de *Smodels* qui est présenté à la section suivante.

Davis et Putnam ont conçu un algorithme pour la résolution de ce problème. Son principe de fonctionnement consiste à énumérer l'ensemble des interprétations d'une formule écrite sous la forme normale conjonctive. L'algorithme qui se présente sous forme de procédure s'interrompt lorsqu'une formule a été trouvée ou lorsque toutes les interprétations ont été énumérées.

Tel que présenté dans [Wur00] cet algorithme se résume tel qu'il suit :

«Soit C un ensemble de clauses et L un ensemble de littéraux. On définit $(C)_L$ en deux étapes de la façon suivante :

1. Retirer de C les clauses contenant au moins un littéral de L.
2. Pour chaque clause restante, retirer chaque littéral dont l'opposé apparaît dans L.

Dans la première étape, on retire les clauses satisfaites par L. Dans la seconde étape, on retire de chaque clause les littéraux falsifiés par L. »

Smodels

Smodels est un système pour la programmation par ensemble de réponses (*ASP*). Il a été développé par I. Niemela et P. Simons [NS97] à la fin des années 1990. Il constitue une implantation de la sémantique des modèles stables pour la programmation logique normale. L'*ASP* est un paradigme de programmation qui diffère de la programmation procédurale par la façon dont les programmes sont décrits. Les programmes sont formés d'atomes et de règles d'inférence. La réponse à un problème se traduit par un ensemble d'atomes qui forment un modèle et ce dernier nous indique quels atomes sont vrais et lesquels sont faux.

Tel que mentionné dans [NS97],

1. *Smodels* calcule le modèle bien formé d'un programme.
2. Il peut déterminer s'il possède un modèle stable.
3. Il peut générer tous les modèles stables d'un programme ou une partie de ceux-ci.
4. Il peut intégrer deux types de requêtes élémentaires. Ainsi, il permet de déterminer si un littéral est vérifié dans un ou dans tous les modèles stables propres à un programme.

Pour parvenir à calculer les modèles stables, lesquels sont des ensembles de littéraux équivalent à la clôture déductive d'un programme logique associé aux ensembles de littéraux respectifs, d'un programme logique normal, un programme logique dans lequel il ne figure pas de symbole de négation classique ¬, deux grandes étapes doivent être parcourues. Ces étapes sont la traduction du programme logique normal en un format compréhensible par *Smodels* à l'aide du logiciel *Lparse* et le calcul des modèles stables proprement dit avec *Smodels*. Ainsi, *Lparse* effectue la lecture des expressions logiques en ASP contenues dans un programme logique normal qui sont enmagasinées dans un fichier texte. Il analyse ces dernières et les décomposent pour ensuite les rendre utilisables par *Smodels*. L'image ci-dessous présente ces deux grandes étapes :

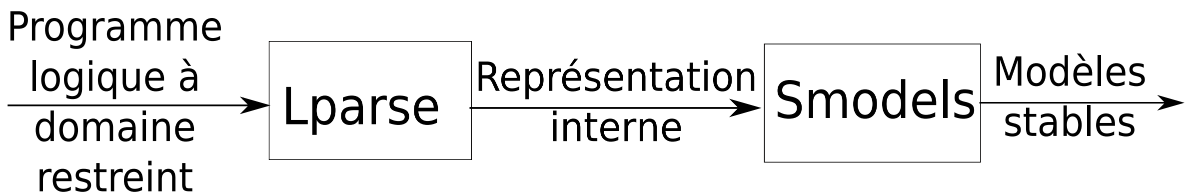

Fig. 2.2 – Traduction d'un programme logique normal en un format compréhensible par *Smodels*

Un programme logique restreint est un programme logique dont le domaine est fini. *Smodels* est à la fois un paradigme de programmation et un outil dont la prédilection est le calcul de modèles stables. Lorsqu'un programme logique normal est soumis à *Smodels*, ce dernier calcule l'ensemble des modèles stables en recourant à un certain nombre d'heuristiques de bases. Dans plusieurs situations, le calcul des modèles stables demeure une opération passablement complexe et peut devenir rapidement une tâche impossible si le problème traité est d'un niveau de complexité calculatoire trop élevé.

Pour ces raisons, Würbel et Papini et Bennaim(2005) [BBJ+05] ont développé des versions modifiées de *Smodels* incluant des heuristiques particulières. Ces méthodes permettent à la fois d'accélérer le calcul de modèles et également d'implanter une méthode de révision des croyances reposant sur le calcul de r-ensembles. Toutefois, pour ce mémoire, nous nous en tenons à présenter *Smodels* dans sa forme originale tel que construit par Niemela [NS97].

L'algorithme de Smodels est représenté par la fonction *Smodels* exposée ci-dessous. La fonction *conflits* dans le programme logique permet de détecter les conflits à l'intérieur du programme. La fonction *expand* calcule l'ensemble des conséquences logiques à partir de l'ensemble d'atomes L.

Voici l'algorithme résumant le fonctionnement de *Smodels* dans sa version originale :

Données: P un programme logique normal fini et L un ensemble fini de littéraux.
Variables du programme : L un ensemble de littéraux et X un atome.
début
  $L = expand(P, L)$.
  si $conflit(P, L)$ **alors**
    retourner FAUX.
  sinon
    si *tous les atomes de* P *sont définis dans* L **alors**
      retourner $test(P, L)$.
    sinon
      Choisir X un atome non couvert par L. **si** $smodels\,(P, L \cup \{notX\})$
      alors
        retourner VRAI
      sinon
        retourner $smodels\,(P, L \cup \{X\})$
      fin
    fin
  fin
fin

2.3 Définition d'une base de croyances

À la lumière de la définition de base de connaissances proposée par Schreiber [SAA+00] que nous considérons dans un contexte évolutif, une base de croyances est un ensemble d'instances de types de croyances présentes dans un schéma de domaine tels que des concepts, des relations, et des règles exprimées à l'aide d'expressions logiques. Un schéma de domaine est un schéma qui représente le domaine à l'étude, soit le « monde des possibles ». Nous appliquerons cette définition à celle de base de géocroyances dans le chapitre 3 de ce mémoire.

2.4 Raisonnement

Les activités de raisonnement s'intègrent à la vie courante de tous les humains sur Terre dans toutes ses activités. Selon le dictionnaire Robert (2001) [Rob94], Il se définit comme une "Suite de propositions liées les unes aux autres selon des principes déterminés, et aboutissant à une conclusion." Dans la section qui suit, nous nous intéresserons plus spécialement aux types de raisonnement logique classique et non monotone.

2.4.1 Le raisonnement classique ou monotone

Le raisonnement classique ou monotone est un type de raisonnement qui suit la propriété de monotonie.

De manière formelle, la propriété de monotonie s'énonce comme suit :

Soient $P_1, ..., P_n$, R, Q des ensembles de formules bien formées.

$\forall P_1$, ..., P_n, R, Q, si de $(P_1, ..., P_n)$ Q peut être déduit, alors

de $(P_1, ..., P_n, R)$ il est toujours possible de déduire Q.

Pour la présentation du raisonnement classique, nous nous basons sur le chapitre 5 du livre «L'espace et le temps en intelligence artificielle : modèles de raisonnement et applications (à paraître)» [GLP07].

Voici quelques définitions utiles pour le raisonnement logique :

Définition 2.36. *Soit F une formule de logique classique $\mathcal{L}$ et une interprétation I, F est valide si et seulement si $\forall I$, $I(F) = 1$.*

Définition 2.37. *Soit F une formule de logique classique, $\mathcal{L}$ et une interprétation I, F est insatisfaisable ou incohérent si et seulement si il n'existe pas d'interprétation I telle que, $I(F) = 1$.*

Cette dernière définition s'applique également à des ensembles de formules tel que dans la définition suivante :

Définition 2.38. *Soit F un ensemble de formules de $\mathcal{L}$, et une interprétation I, F est insatisfaisable si et seulement si il n'existe pas d'interprétation I telle que pour toute formule F , $F \in F$, $I(F) = 1$.*

Les méthodes présentées dans ce qui suit sont des méthodes de raisonnement par l'absurde. Ainsi, il est prouvé que la négation d'une formule est incohérente. Le raisonnement déductif présenté repose sur le théorème suivant :

Théorème Soit $F \in L$, $F \models F$ si et seulement si $F \cup \{\neg F\}$ est insatisfaisable ou incohérent.

Ce théorème permet de ramener le théorème de la conséquence logique à celui de la satisfaisabilité.

Résolution d'ensembles de clauses en logique propositionnelle

La méthode de résolution proposée dans [Rob79] consiste à tester l'incohérence d'un ensemble de clauses et à vérifier si cet ensemble contient la clause vide. S'il la contient, il découle que l'ensemble est cohérent. Dans le cas contraire, il faut vérifier si la clause vide peut être une conséquence logique de cet ensemble de clauses.

De façon non formelle, ceci revient au problème suivant. Soit A, B, X des formules propositionnelles. On suppose l'existence d'une interprétation qui vérifie $(A \vee X)$ et $(B \vee \neg X)$. S'il existe une interprétation qui satisfait X alors il suit que B est satisfait. Si l'interprétation ne satisfait pas X, alors elle vérifie A.

Exprimé de manière formelle ceci revient à :

$\{A \vee X, B \vee \neg X\} \models A \vee B$

dans laquelle X est une proposition et A et B sont des clauses.

Règle de résolution : Soient S un ensemble de clauses, c_1 et c_2 des clauses de S et l un littéral. Si l est présent dans c_1 et $\neg l$ est présente dans c_2, $C_1 = l_{i1} \vee l_{i2} \vee ... \vee l_{in} \vee l$ $C_2 = l_{j1} \vee l_{j2} \vee ... \vee l_{jm} \vee \neg l$ il découle que $r = l_{i1} \vee l_{i2} \vee .. \vee l_{in} \vee l_{j1} \vee l_{j2} \vee ... \vee l_{jm}$ constitue une conséquence logique de S et se nomme résolvante.

Dans le but de montrer qu'un ensemble de formules est incohérent, ces dernières formules doivent être exprimées sous la forme conjonctive normale en appliquant le théorème de normalisation qui produit un ensemble de clauses équivalent.

Afin de résoudre un ensemble de clauses, l'algorithme de résolution suivant, tel que présenté dans [GLP07], peut être utilisé :

Données:
 en entrée : S et S' : des ensembles de clauses et la variable *possible*
 en sortie : S' : un ensemble de clauses
début
 $S' \leftarrow S$.
 $possible \leftarrow vrai$
 tant que $\square \notin S'$ *et possible* **faire**
  si *il est possible de produire une nouvelle résolvante* **alors**
   Choisir l, c_1, c_2 tels que $l \in c_1$ et $\neg l \in c_2$
   Calculer la résolvante r
   $S' \leftarrow S' \cup \{r\}$
  fin
  $possible \leftarrow faux$
 fin
fin

note : $\square$ désigne la clause vide.

Résolution d'ensembles de clauses en logique des prédicats

En logique des prédicats, le théorème de Herbrand offre la possibilité de convertir un problème de satisfaisabilité d'un ensemble de formules en un problème de satisfaisabilité d'un ensemble de clauses de la logique propositionnelle. Pour réaliser une telle opération, l'ensemble C des clauses correspondantes est associé à la forme conjonctive normale équivalente à la formule ou à l'ensemble de formules de la logique des prédicats. Ensuite, à partir de ces clauses, un système appelé système de Herbrand est réalisé en instanciant les variables à partir des éléments pris dans l'univers de Herbrand. Cette opération se nomme saturation. Par la suite, il est possible d'appliquer le principe de résolution. Or, les opérations de saturation et de résolution sont commutatives tel qu'il a été démontré par A. Robinson dans [Rob79]. Tel que rapporté par Papini dans [GLP07], débuter avec la résolution avant l'instanciation dans l'univers de Herbrand comporte l'avantage d'amenuiser la complexité algorithmique en opérant sur des paires de clauses qui sont présentes en quantités finies et contiennent un nombre fini de littéraux.

Pour être en mesure d'appliquer le principe de résolution en premier, il est nécessaire d'effectuer une unification dont le but visé est de rendre possible la coïncidence des atomes par un choix stratégique de termes qui remplacent les variables. Voici quelques définitions essentielles à la présentation de l'algorithme d'unification qui suit :

Définition 2.39. ***Substitution*** *: Application notée σ partant de l'ensemble des variables vers l'ensemble des termes qui est quasiment partout égale à l'identité.*

Définition 2.40. ***Substitution de renommage*** *: Substitution constituant une bijection sur l'ensemble des variables.*

Définition 2.41. ***Instance*** *: Une instance d'un terme t, se note $\sigma(t)$ et est obtenue en remplaçant toutes les occurences de la variable x par $\sigma(x)$. Soit t_1 et t_2, t_2 constitue une instance de t_1 s'il existe une substitution σ telle que $t_2 = \sigma(t_1)$.*

Définition 2.42. ***Unifiabilité*** *: Deux termes t et t' sont dits unifiables s'il existe une substitution σ telle que $\sigma(t) = \sigma(t')$. Dans ce contexte particulier, la substitution se nomme l'unificateur.*

Définition 2.43. ***Unificateur général*** *: Un unificateur σ' est plus général que σ, s'il existe une substitution σ'' telle que $\sigma'' \circ \sigma'$ où $\circ$ désigne la composition de deux unificateurs.*

Il est à noter que l'unification n'est possible qu'entre une variable et un terme.

La résolution en logique des prédicats repose sur le théorème suivant :

THÉORÈME DE RÉSOLUTION. *Soit S un ensemble de clauses, c_1 et c_2 des clauses de S, $l_1 \in c_1$ et $\neg l_2 \in c_2$. Définissons θ en tant que substitution de renommage de manière à ce que $\theta(c_1)$ et c_2 ne possèdent aucune variable libre commune, σ l'unificateur principal de $\theta(c_1)$ et c_2 et la clause de résolution nommée résolvante $r = \sigma(\theta(c_1 \setminus \{l_1\}) \vee (c_2 \setminus \{\neg l_2\}))$ alors il découle que S et $S \cup \{r\}$ sont logiquement équivalents.*

Nous sommes maintenant en mesure de présenter l'algorithme d'unification tel que présenté dans [GLP07] qui est le suivant :

algorithme unification(t_1, t_2 sont des termes) :($possible$, σ')
Données: n, m, i, x, t, σ', $possible$
début

```
    si l'un des termes t1 ou t2 est une variable x et l'autre est t alors
        si x = t alors
            possible ← vrai
            σ ← ∅
        sinon
            si x apparaît dans t alors
                possible ← faux
            sinon
                possible ← vrai
                σ ← (σ(x) = t)
            fin
        fin
    sinon
        (t1 = f(x1, ..., xn) et t2 = g(y1, ..., ym))
        si f ≠ g ou n ≠ m alors
            possible ← faux
        sinon
            i ← 0
            possible ← vrai
            σ ← ∅
            tant que i < n et possible faire
                i ← i + 1
                (possible, σ') ← unification(σ(xi), σ(yi))
            fin
            si possible alors
                σ ← σ' ∘ σ
            fin
        fin
    fin
```

fin

Voici l'agorithme de résolution pour la logique des prédicats tel que présenté dans [GLP07] :

Données:
  en entrée : S et S' : des ensembles de clauses et la variable *possible*
  en sortie : S' : un ensemble de clauses

début
  $S' \leftarrow S$.
  $possible \leftarrow vrai$
  tant que $\Box \notin S'$ *et possible* **faire**
    si *il est possible de produire une nouvelle résolvante* **alors**
      Choisir l_1, l_2, c_1, c_2 tels que $l \in c_1$ et $\neg l \in c_2$ et l_1, l_2 unifiables
      Calculer la résolvante r à l'aide de l'unificateur principal
      $S' \leftarrow S' \cup \{r\}$
    fin
    $possible \leftarrow faux$
  fin
fin

Le raisonnement classique ne permet pas d'effectuer de révisions. Or, en géomatique, les besoins de révisions de croyances sont très fréquents étant donné que les informations spatiales évoluent considérablement au fil du temps dans la majorité des contextes. Pour cet raison, le raisonnement classique se prête souvent mal au raisonnement sur des informations spatiales dans des contextes évolutifs. Ainsi, c'est, entre autres, pour cette raison que le raisonnement non monotone est d'un vif intérêt en géomatique.

Une des difficultés que pose le raisonnement classique, dans le contexte de la logique propositionnelle ou des prédicats, est celle du traitement des contradictions. Ceci est attribuable à l'égalité de l'importance de toutes les expressions logiques. Ainsi, il n'est pas possible d'introduire des expressions logiques qui auront préséance sur d'autres, ce qui revient à affirmer que le raisonnement sur de l'information partiellement ordonnée n'est pas possible [Sch97]. C'est dans ce genre de situation que le raisonnement non monotone présente tout son intérêt.

2.4.2 Le raisonnement non monotone

Le raisonnement non monotone tire ses origines historiques dans la tentative des humains de modéliser le raisonnement reposant sur le sens commun. Les conclusions déduites à la lumière d'un tel type de raisonnement ne sont pas absolues dans tous les cas. Le raisonnement non monotone nous permet de recourir à des stratégies de raisonnement qui reposent sur des heuristiques qui permettront de parvenir à tirer des conclusions plus rapidement dans certains contextes.

Le raisonnement non monotone implique généralement le recours aux logiques non monotones. Les premières logiques non monotones sont apparues vers le début des années 1980. Elles ont été introduites par Reiter, McDermott et Doyle afin de répondre à des besoins spécifiques de raisonnement en intelligence artificielle. Ces logiques possèdent la particularité que lors de l'ajout d'une prémisse, ce type de raisonnement peut nous conduire à retirer une conclusion tirée précédemment. Ainsi une conclusion dans ce cas constitue une hypothèse, et ce, jusqu'à la preuve du contraire. Ces conclusions constituent des inférences par défaut.

La propriété de non-monotonie propre au raisonnement non monotone s'exprime de la façon suivante :

$\forall P_1$, ..., P_n, R, Q, si de P_1, ..., P_n on peut déduire Q alors

de P_1, ..., P_n, R on ne peut plus déduire Q.

Dans le cadre de ce travail nous utilisons une approche de raisonnement non monotone à travers la méthode de révision des r-ensembles. Toutefois, avant de détailler cette méthode, nous présentons une section sur le raisonnement spatial afin de la situer dans son contexte d'application, et nous détaillons ce en quoi consiste la révision des croyances sur laquelle repose la méthode des r-ensembles.

2.5 Raisonnement spatial

Dans notre contexte d'étude, le raisonnement spatial prendra le sens de raisonnement sur l'espace physique à partir de supports de représentations logiques tels que ceux qu'offrent les langages de programmation *PROLOG* et *ASP*. Ainsi, nous serons amenés à effectuer des raisonnements logiques à l'intérieur de bases de croyances. Les interprétations que nous pouvons tirer sur l'espace physique peuvent être falsifiées si notre représentation de l'espace physique comprise à l'intérieur d'une base de géocroyances n'est pas cohérente. Dans de telles situations, effectuer une révision des croyances devient une activité très utile, ce que nous verrons dans la section qui suit.

Cohn et Egenhofer ont apporté des contributions importantes dans le domaine du raisonnement spatial en proposant des modèles topologiques pour ce type de raisonnement. Ainsi, dans ce domaine, Cohn a développé le RCC8 [Coh96], [RCC92] un calcul relationnel permettant de représenter l'information spatiale et topologique.

Le schéma 2.4, issu de [Coh96], présente les différentes relations spatiales de base proposées par Cohn dans le RCC8 ainsi que les transitions existant entre ces relations :

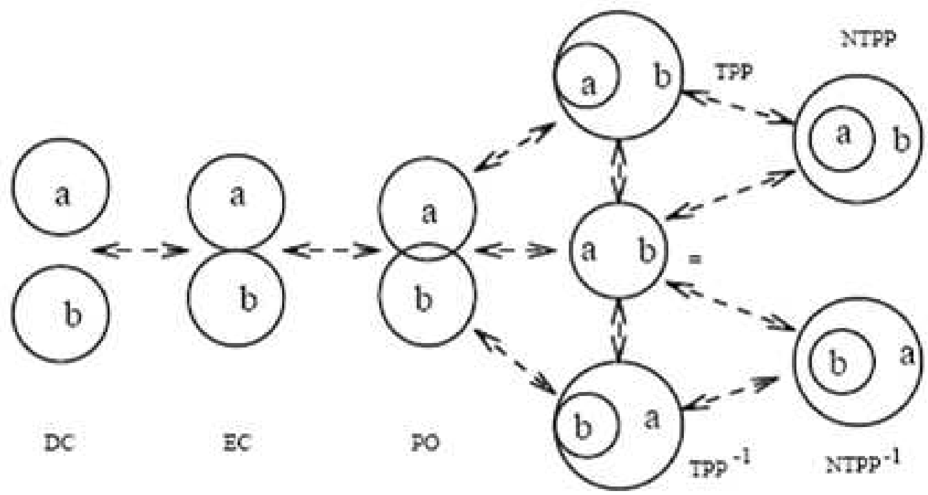

Fig. 2.3 – Le modèle de calcul relationnel RCC8 de Cohn

Ce schéma résume les différentes relations spatiales entre deux objets spatiaux a et b caractérisées par le modèle de calcul relationnel RCC8. DC signifie "déconnecté de", EC "est connecté extérieurement à", PO "recouvre partiellement", TPP "est intérieur et tangentiel à", TPP^{-1} "est intérieur et tangentiel à" sous sa forme réciproque, $NTPP$ "est intérieur et non tangentiel à", $NTPP^{-1}$ sous sa forme réciproque.

Parallèlement, sensiblement dans les mêmes années, Egenhofer a développé le modèle des 9 intersections [EH90]. Ce modèle (FIG 2.4) permet de caractériser les différentes façons suivant lesquelles des objets peuvent entrer en relations topologiques entre eux. Son modèle permet de décrire les relations topologiques pouvant exister dans des objets géographiques et d'avoir une bonne compréhension des relations géométriques qui existent entre de tels objets.

Ce modèle peut s'exprimer à l'aide du système matriciel suivant proposé par Egenhofer :

$$R(A,B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

Fig. 2.4 – Le modèle des 9 intersections de Egenhofer représenté sous forme de système matriciel

Dans la figure 2.4 issue de [EH90], $R(A, B)$ désigne l'ensemble des neuf relations topologiques qui lient A et B dans le modèle des neuf intersections de Egenhofer. A° désigne l'intérieur de l'objet A, la frontière de A est représentée par ∂A et l'extérieur de A se note A^-. De la même façon, B° désigne l'intérieur de l'objet B, la frontière de B est représentée par ∂B et l'extérieur de B se note B^-.

Les formalismes de représentation de l'espace de Egenhofer et de Cohn même s'ils n'ont pas été utilisés directement lors de la création de la BNDT pourraient très bien servir à reformuler les différents types de relations spatiales représentés dans la BNDT afin d'enrichir des modèles de connaissances tel que celui qui est présenté dans le troisième chapitre de ce mémoire. Ces deux modèles constituent aujourd'hui des références pour le raisonnement spatial et sont des représentations très flexibles de relations spatiales, ce qui peut présenter des avantages non négligeables s'ils sont intégrés aux modèles de connaissances. Ces avantages seraient perceptibles lors d'analyses spatiales ou lors de différentes formes d'opérations calculatoires visant la vérification de cohérence de bases de croyances élaborées à partir de modèles de connaissances.

2.6 Révision des croyances

Qu'est-ce que la révision ?

La révision est une opération qui vise à permettre la préservation ou la restauration de la cohérence à l'intérieur d'une base de croyances K lors de l'ajout de nouvelles informations. Une révision constitue une actualisation des croyances s'accomplissant dans un monde statique. Elle est très utile pour permettre le maintien ou le rétablissement de la cohérence à l'intérieur d'une base de géocroyances.

La révision s'intéresse à la façon selon laquelle un agent modifie ses croyances en intégrant de nouvelles croyances dans un monde statique. Le principal objectif poursuivi par l'opération de révision est de prôner l'amélioration de la qualité d'ensemble des croyances portant sur une situation de nature statique et imparfaitement appréhendée initialement et de voir au maintien de la cohérence entre les croyances dans une base de croyances.

La théorie de la révision est née d'un problème de raisonnement propre au monde légal. Un exemple de problème typique est celui de l'instauration de nouvelles lois qui, quelquefois, entrent en contradiction avec les contraintes liées à un ancien corps de lois. Dans une telle situation, un problème d'information conflictuelle apparaît. Normalement, une nouvelle loi devrait prévaloir sur des anciennes lois qui la contredisent, à l'exception du cas où si il est prévu par principe que les anciennes lois prévalent sur la nouvelle loi. Or, l'avènement de la présence d'informations conflictuelles n'entraîne pas nécessairement que toutes les anciennes lois deviennent invalides. La théorie de la révision nous conduit à nous intéresser à la partie conflictuelle et à identifier comment la cohérence peut être rétablie le plus simplement possible afin de mettre fin au conflit.

Le processus de révision implique généralement deux grandes étapes soit le retrait ou l'ajout d'un minimum d'information d'une base de croyances, et l'union de la base de croyances altérée, de manière à ce que l'information conflictuelle de la base de croyances initiale ait été éliminée, avec l'ajout de la nouvelle information.

Plusieurs méthodes de révision existent, parmi celles-ci se trouvent, la révision selon Spohn [Spo88], la révision itérée [DP97], la révision bayesienne [Pea88], la révision selon Dalal [Dal88], la révision avec les r-ensembles [Pap00] et plusieurs autres. Pour les fins de ce travail, nous avons retenu la révision avec les r-ensembles qui repose sur le cadre d'Alchourron, Gardenfors et Makinson (AGM) parce qu'elle s'adapte bien au contexte de la BNDT et permet d'éliminer efficacement les incohérences qu'elle contient.

Dans la section qui suit, nous présentons le cadre de révision AGM pour des théories, un cadre qui s'avère très utile pour la révision des croyances et qui se révèle bien connu chez les scientifiques qui travaillent sur le problème de la révision des croyances.

2.6.1 Le cadre d'Alchourron, Gardenfors et Makinson (AGM)

Trois principes ou idées se situent à la base du cadre ou paradigme AGM pour la révision : 1) Le principe de la cohérence : Une opération de révision doit conduire à une base de croyances cohérente. 2) Le principe du changement minimal : Un minimum de croyances devrait être modifié par l'opération de révision. 3) Le principe de priorité : Les nouvelles croyances prévalent sur les anciennes croyances.

Ces idées sont exprimées par les AGM à travers des postulats que nous présentons dans la présente section. Ces idées se situent également à la base de la méthode des r-ensembles détaillée plus loin dans ce chapitre.

Les postulats AGM portent sur des théories. Une théorie est un ensemble déductivement clos et infini. Dans un premier temps AGM [AGM85] ont suggéré des postulats stipulant quel doit être le «bon» comportement d'un opérateur de révision. Un opérateur de révision sert à spécifier les règles qui dictent la façon de réviser les croyances pour rétablir la cohérence à l'intérieur d'une base de croyances. Dans un deuxième temps, ils ont proposé de définir un opérateur de révision à partir d'un opérateur de contraction tel qu'expliqué dans [Pap00]. L'opération de contraction se note $T - A$. Elle vise à créer une théorie T' telle que $T' \cup \{\neg\phi\}$ est cohérent. La révision se note $T * A$ et elle consiste à prendre une théorie T et une formule A et à introduire A dans T de manière à préserver la cohérence.

Dans l'objectif de parvenir à réaliser ces deux opérations, AGM proposent les postulats suivants tels que présentés par Papini et Würbel dans [Wur00] :

$(G * 1)$ $T * A$ est une théorie. Une théorie révisée par une formule est également une théorie.

$(G * 2)$ $A \in T * A$. A est élément de la théorie révisée.

$(G * 3)$ $T * A \subseteq T + A$. La théorie révisée est un sous-ensemble de l'expansion d'une théorie T avec la formule A si $A-$.

$(G * 4)$ Si $\neg A \notin T$ alors $T * A = T + A$. Si $\neg A$ n'est pas inclus dans T, il suit que la révision de la théorie T par la formule A est équivalente à l'expansion de T et A.

$(G * 5)$ $T * A = T^{\perp}$ seulement si A est insatisfaisable. Ce postulat garantit que l'opération $T * A$ génère un ensemble cohérent de formules mis à part si A constitue une antilogie et ainsi ne peut être vérifié.

$(G*6)$ Si $A \equiv B$ alors $T*A = T*B$. Le résultat de la révision doit être indépendant de la forme syntaxique de la formule ajoutée.

$(G*7)$ $T*(A \wedge B) \subseteq (T*A)+B$. La révision de T par A et B est incluse dans la théorie T révisée par A à laquelle B est ajoutée.

$(G*8)$ Si $\neg B \notin T*A$ alors $(T*A)+B = T*(A \wedge B)$. Au minimum, réviser T pour inclure A et B devrait réduire la révision à une expansion de T par A, dans la mesure où B ne contredit pas $T*A$.

La fermeture déductive de $T-A$ constitue un sous-ensemble de la fermeture déductive de T. À la lumière de ces postulats, en utilisant les identités de Levi, soit $K*A = (K-\neg A)+A$, et de Harper, soit $K-A = K \cap (K*\neg A)$, tel qu'expliqué dans [Pap00] il suit que l'opération de révision peut se traduire par $T*A := (T-(\neg A)) \cup \{A\}$ et que celle de contraction peut être exprimée par $T-A := T \cap (T*(\neg A))$.

Afin de traiter le problème relié à l'indétermination du changement minimal, AGM ont introduit la notion d'ordre partiel sur les formules dans des langages donnés. Ceci prend le nom de priorité épistémique [Sch97]. D'après AGM, $\phi \vdash \psi$ implique $\phi \leq \psi$ ce qui revient à affirmer que si ψ est une conséquence logique de ϕ alors nous croyons au moins autant en ψ qu'en ϕ.

Pour ce travail, nous avons retenu la méthode de révision avec r-ensembles [Pap92], qui suit les trois principales idées à la base du paradigme AGM, car elle permet de déceler efficacement les incohérences et de rétablir la cohérence de façon stratégique, ce qui convient bien pour notre problème de recherche. Il existe également une implantation de cette méthode de révision qui est adaptable à l'échantillon de base de géocroyances que nous utilisons pour ce travail de recherche.

Avant de décrire cette méthode, il importe de définir ce que nous entendons par *état épistémique* d'un agent dans le contexte de la représentation de croyances évolutives. Un état épistémique représente les croyances qu'un agent possède sur le monde réel en s'appuyant sur les informations qui lui sont disponibles.

L'opérateur * est un opérateur de révision. Ainsi, l'expression $T * A$, signifie que la théorie T est révisée par l'ensemble A.

Dans ce qui suit, nous présentons la méthode des r-ensembles, la construction d'un programme logique et la recherche de modèles stables essentiellement à la lumière de l'article «An Answer Set Programming Encoding of Prioritized Removed Sets Revision :

Application to GIS» [BNBPW04]. Dans ce contexte, les postulats des AGM ne sont plus appliqués à des théories qui sont infinies mais bien à des bases de croyances qui, elles, sont finies.

Méthode des r-ensembles

La méthode des r-ensembles est une approche de révision basée sur la cardinalité d'ensembles.

La variable Ψ est définie en tant qu'état épistémique et l'expression $Bel(\Psi)$ en tant qu'ensemble de croyances formé de formules propositionnelles. Considérant que toute formule propositionnelle équivaut à sa forme conjonctive normale correspondante, il suit que $Mod(Bel(\Psi)) \equiv Mod(K)$. La dernière expression logique se lit «Un modèle de $Bel(\Psi)$ est équivalent à un modèle de la base de croyances K». Dans le présent cas d'application géomatique ou encore dans des contextes informatiques, les bases de croyances traitées sont des ensembles finis de clauses.

La présente approche de révision vise à identifier les plus petits ensembles de clauses incohérents et à les retirer d'une base de croyances afin de rétablir la cohérence de cette dernière. Le théorème de compacité assure l'existence d'au moins un sous-ensemble incohérent dans une base de croyances. L'approche de révision consiste à déterminer quels ensembles minimaux de formules intersectent avec les ensembles minimaux incohérents. Ainsi, les r-ensembles constituent des ensembles minimaux incohérents de clauses pouvant être retirés afin de rétablir la cohérence d'une base de croyances.

En termes formels, soit K une base de croyances et A des ensembles de clauses tels que $K \cup A$ est incohérent, dans laquelle expression $\cup$ est un opérateur d'union d'ensembles. Un r-ensemble se définit comme suit :

Définition 2.44. *Soient K et A deux ensembles cohérents de clauses tels que $K \cup A$ est incohérent. R formé d'un sous-ensemble de K, est un r-ensemble de $K \cup A$ si et seulement si :*

1) $(K \backslash R) \cup A$ est cohérent ;
2) $\forall R' \subseteq K$, si $(K \backslash R') \cup A$ est cohérent alors $\mid R \mid \leqslant \mid R' \mid$.

note : Dans l'expression $(K \backslash R') \cup A$, $\backslash$ signifie «moins».

L'ensemble des r-ensembles de $(K \cup A)$ se note $\mathcal{R}(K \cup A)$. Les r-ensembles sont tels que si $(K \cup A)$ est incohérent alors $(K \cup A) \neq \emptyset$.

Afin de montrer en détail comment fonctionne cette méthode, nous présentons un exemple d'application à la BNDT de cette approche dans le chapitre 4 de ce mémoire.

La révision suivant l'approche des r-ensembles se définit à l'aide d'une fonction de sélection s qui a pour rôle de choisir un sous ensemble non-vide de $\mathcal{R}(K \cup A)$ si $K \cup A)$ est incohérent et retourne $\emptyset$ si $\mathcal{R}(K \cup A)$ est cohérent.

L'opération de révision par l'approche des r-ensembles (RSR) se définit formellement par :

$$K \circ_{RSR} A =_{def} \bigvee_{R \in s(R(K \cup A))} Cn((K \backslash R) \cup A)$$

La méthode de révision r-ensembles proposée par Papini et Würbel nécessite le calcul des échantillons minimaux obtenus lors de l'application de l'algorithme de Reiter [Rei87]. Elle consiste à calculer les r-ensembles et à les retirer de la base de croyances K.

Plus récemment, cette méthode a été raffinée, améliorée et adaptée pour être implantée dans l'outil *Smodels* présenté précédemment dans ce chapitre.

La méthode des r-ensembles nécessite 3 étapes qui sont :

1) L'introduction de règles de manière à ce que les modèles stables de $P_{K \cup A}$ entrent en bijection avec les interprétations des variables propositionnelles contenues dans V^+.

2) Des règles sont introduites pour empêcher que les interprétations I qui ne sont pas des modèles de A de correspondre à des modèles stables.

3) Les interprétations de I qui correspondent à des modèles stables de $P_{K \cup A}$ et qui sont des modèles de $(K \cup A) \backslash Cls(I \cap R_K^+)$ sont sélectionnées.

Construction d'un programme logique normal

Pour les fins de l'application de la méthode des r-ensembles, il est essentiel de construire un programme logique normal. Un programme logique normal est un programme logique dans lequel ne figure pas le symbole de négation classique $\neg$. Cette étape est importante puisque les algorithmes de la version adaptée de *Smodels* pour la révision ne fonctionnent qu'avec les programmes logiques normaux.

Un programme logique P constitue un ensemble de règles r de la forme :

$l_0 \leftarrow l_1, l_2, ..., l_m,$ *not* $l_n, l_{n+1}, l_{n+2}, ..., l_m$

dans laquelle *not* désigne le symbole de négation par échec. Par exemple, si nous avons l'expression *not* p, dans laquelle p constitue une proposition, cela signifie que nous ignorons si p est vraie et que jusqu'à preuve du contraire, p est supposée fausse.

Soit P un programme logique, K une base de croyances et A un ensemble de clauses avec lequel il faut réviser K. Un programme logique normal P_{KUA} doit être construit de manière à ce que les modèles stables propres à ce programme correspondent aux sous-ensembles R de K tels que $(KUA)\backslash R$ soit cohérent. Dans ce but, les modèles stables préférés qui correspondent aux r-ensembles de KUA nécessitent d'être calculés. Les modèles stables préférés sont des modèles qui correspondent aux plus petits ensembles de clauses à retirer de KUA afin de restaurer la cohérence au sein de la base de croyances K.

Afin de parvenir à construire un programme logique convenant à l'application de la méthode des r-ensembles, deux applications intermédiaires sont nécessaires. La première application *Clause* Cls de R_K^+ dans K, qui sert à récupérer une clause c à partir d'une règle r_c et la deuxième *Inverse* Inv de V dans $Lit(V^+)$ sert à passer d'un atome artificiel a' à son expression originale avec négation classique $\neg a$.

Les applications Cls et Inv sont définies telles que :

$Cls : R_K^+ \longrightarrow V$ où $\forall r_c \in R_K^+, Cls(r_c) = c$

$Inv : V \longrightarrow Lit(V^+)$ où

$$\forall a \in V, Inv(a) = \begin{cases} a \text{ si } a \in V^+, \\ \neg b \text{ si } a = b^{'} \text{ et } b^{'} \in V^- \end{cases}$$

Ces deux applications peuvent sembler absconses à prime abord, cependant leurs rôles seront plus faciles à saisir lorsqu'elles seront mises en application dans le chapitre 4 dans l'exemple d'application de la révision des géocroyances avec la méthode des r-ensembles.

Le programme initial ne contient que la clause vide. Ainsi, $P_{KUA} = \varphi$. Nous introduisons les différentes règles du programme en suivant 3 étapes qui sont les suivantes :

Nous visons à réviser K par A. Voici les trois étapes de construction du programme logique $P_{K \cup A}$ afin d'appliquer la méthode de calcul des r-ensembles pour la révision :

1)Nous introduisions des règles de manière à ce que les modèles stables de $P_{K \cup A}$ entrent en bijection avec les interprétations de V^+. Ainsi, $\forall a$ de V^+, afin de prendre en compte la négation classique de certains atomes, nous introduisons les règles $a \leftarrow not\ a'$ et $a' \leftarrow not\ a$.

2) Nous introduisons des règles pour empêcher que les interprétations I qui ne sont pas des modèles de A de correspondre à des modèles stables. Ainsi, pour toute clause c de A telle que c= $\neg a_0 \vee ... \vee \neg a_n, a_{n+1}, ..., a_m$, nous ajoutons la règle suivante :

$false \leftarrow a_0, ..., a_n, a'_{n+1}, ..., a'_m$

Au programme, pour éliminer l'atome *false* des modèles stables recherchés, nous ajoutons la règle :

$contradiction \leftarrow false, not\ contradiction$

ce qui permet de faire en sorte que les modèles stables de $P_{K \cup A}$ correspondent aux interprétations de V^+ qui sont des modèles de A.

3) Nous sélectionnons les interprétations de I qui correspondent à des modèles stables de $P_{K \cup A}$ et qui sont des modèles de $(K \cup A) \setminus Cls(I \cap R_K^+)$. Pour ce faire, pour chaque clause c de K tel que $c = \neg b_0 \vee ... \vee \neg b_n, b_{n+1}, ..., b_m$, nous introduisons la règle suivante : $r_c \leftarrow b_0, ..., b_n, b'_{n+1}, ..., b'_m$.

Recherche de modèles stables

La recherche de modèles stables propres à un programme logique nécessite la réduction de ce programme. La réduction du programme Δ par rapport à un sous-ensemble de littéraux X appartenant au programme Δ se calcule en retirant de ce programme :

1. Chaque règle r pour laquelle $l \in X$ et $notl \in corps^-(r)$
2. Tous les éléments appartenant à $corps^-$ pour l'ensemble des autres règles

Cette dernière opération de réduction revient à supposer que X est un modèle stable potentiel et à transformer le programme Δ en programme basique de manière à ce que autant du programme initial que du programme réduit il soit possible de déduire le sous-ensemble X. Ceci s'exprime de façon formelle par : $\Delta \models X \Longleftrightarrow \Delta^X \models X$

On affirmera que X est un modèle stable de Δ si et seulement si X est égal à la clôture déduite de Δ^X. i.e. $X = Cn(\Delta^X)$.

Ainsi, pour vérifier si un sous-ensemble X est un modèle stable, il faut simplement recourir à la méthode du point fixe pour évaluer $Cn(\Delta)$ et comparer sa valeur à celle de X. S'il y a une coïncidence des valeurs respectives, il suit que X constitue un modèle stable de Δ. Lorsque les programmes logiques deviennent complexes, afin de réaliser des économies de temps importantes, il devient intéressant d'utiliser un outil tel que Smodels [NS97] pour effectuer la réduction de programmes logiques et le calcul de modèles stables.

Correspondance entre r-ensembles et modèles stables préférés

Un modèle stable préféré est un modèle stable qui conduit directement à un r-ensemble.

Dans cette section, nous montrons l'équivalence qui existe entre les r-ensembles et les modèles stables préférés pour un programme P_k à l'aide de deux propositions pour des bases de croyances non stratifiées.

Définition modèle préféré : Soit S un ensemble d'atomes. S constitue un modèle stable préféré de $P_{K\cup A}$ si et seulement si S est un modèle stable de $P_{K\cup A}$ et que $\forall S'$ modèle stable de $P_{K\cup A}$, S' n'est pas préféré à S.

La correspondance entre les modèles préférés et les r-ensembles s'exprime à l'aide de la proposition suivante :

Proposition : Soit S_p l'ensemble de tous les modèles préférés de $P_{K\cup A}$ et $\mathcal{R}_p$ de tous les r-ensembles appartenant à $K \cup A$.

$$\left\{Cls(S \cap R_K^+) \mid S \in S_p\right\} = \mathcal{R}_p.$$

Cette définition et cette proposition permettent de saisir clairement l'équivalence qui existe entre les notions de r-ensembles et de modèles stables préférés.

2.6.2 Mise à jour des croyances

La mise à jour s'intéresse à la méthode à emprunter par un agent pour prendre en compte que le monde a changé. Une mise à jour constitue une actualisation des croyances s'effectuant dans un monde dynamique. Un agent est un programme qui accomplit des tâches comme un automate et suivant les volontés de son auteur.

2.6.3 Distinctions entre les opérations de mise à jour et de révision

Voici un exemple de situation explicitant bien la différence entre une mise à jour et une révision :

Un agent de parc sait que dans un parc il y a exclusivement soit des érables, soit des bouleaux. Supposons alors que l'agent apprenne alors qu'il n'y a pas d'érables dans la forêt. L'agent, en révisant ses croyances, peut déduire que la forêt contient des bouleaux. Dans ce cas, l'agent a effectué une révision de ses croyances. Ainsi, une modification de ses croyances entraînée par l'ajout de nouvelles informations permet à l'agent d'améliorer la qualité de ses croyances au sujet d'une situation statique et imprécisément appréhendée initialement.

Supposons maintenant que nous ne sommes plus dans la situation d'exclusivité décrite dans le paragraphe précédent. Il peut donc y avoir simultanément des bouleaux et des érables ou ne pas y avoir de ces espèces d'arbres. Nous ne sommes donc plus dans une situation qui est statique. Supposons cette fois qu'il y avait des érables dans la forêt mais que l'agent apprend que quelqu'un les a coupés. Dans cette situation, il devient impossible pour l'agent de conclure qu'il y a des bouleaux dans la forêt. Il est seulement en mesure d'affirmer qu'il n'y a pas d'érables dans la forêt. Dans ce cas, l'agent effectue une mise à jour. Il est conduit à s'intéresser aux explications concernant les changements entraînés par l'apparition de nouvelles informations dans une situation dynamique.

2.7 Bilan de chapitre

Au cours de ce chapitre, nous avons présenté les fondements de la théorie sur les ontologies en géomatique. Nous avons expliqué les différences fondamentales qui existent entre le raisonnement monotone et le raisonnement non monotone dans le contexte spatial. Nous avons présenté des théories qui circonscrivent les problèmes de vérification de cohérence et de révision de géocroyances. Nous avons présenté différents formalismes de programmation logique tels que *PROLOG* et l'*ASP* et une mise en oeuvre particulière du calcul des modèles stables (ou ensembles-réponses), soit le système *Smodels*. Ces derniers permettent de traiter des problèmes dans des contextes pratiques de raisonnement spatial. Dans le chapitre suivant, nous verrons plus en détail à quoi se rattachent toutes ces notions.

Chapitre 3

Définition d'une base de géocroyances

Dans ce chapitre, nous présentons les bases de connaissances telles qu'elles existent actuellement en géomatique. Afin, de les situer en fonction d'une perspective plus large, nous recourons à des notions fondamentales d'ingénierie des connaissances tels que les modèles de connaissances pour former des bases de croyances que nous concevons en utilisant de l'information à caractère géographique. Ceci nous amène à introduire le concept de bases de géocroyances pour des fins de raisonnement monotone et non monotone.

3.1 Les bases de connaissances en géomatique

Les bases de connaissances en géomatique constituent essentiellement des bases de connaissances dont le contenu est à caractère géographique. Elles présentent un intérêt particulier parce qu'elles sont un outil procurant un support représentationnel permettant de raisonner sur des connaissances géographiques. Dans «A Formal Method for Quality Assesment of Spatial Data Bases : The Role of Ontologies», Mostafavi et Edwards [MEJ04a] définissent une base de connaissances comme une représentation d'une ontologie spatiale en *PROLOG*, ce qui constitue une représentation formelle d'une telle ontologie. Le *PROLOG* est un langage de programmation logique reposant sur la logique du premier ordre. Pour les fins de leurs travaux, ils s'appuient sur l'ontologie de la Base nationale de données topographiques du Canada (BNDT). Celle-ci sert de cas d'étude pour une méthode de vérification de cohérence réalisée à l'aide d'un moteur de raisonnement de *PROLOG*. Dans la figure 3.1, nous voyons un résumé des étapes impliquées dans l'étude de la cohérence des données versus son ontologie et à quel niveau la base de connaissance est impliquée dans cette étude.

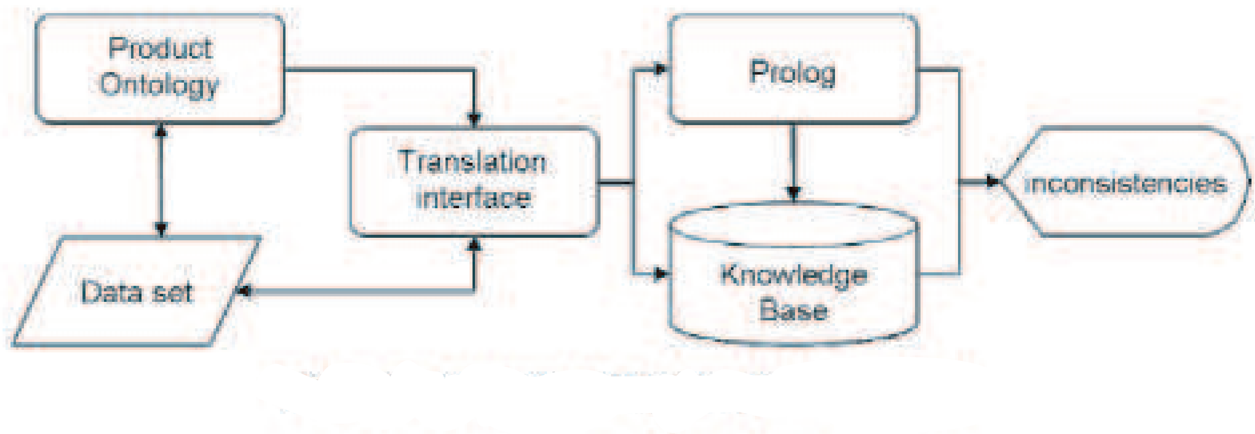

Fig. 3.1 – Étude de la cohérence d'une base de données spatiales avec son ontologie [MEJ04a]

Dans la figure 3.1, nous pouvons voir que dans un premier temps, une ontologie géographique est définie, et parallèlement, un ensemble de données est créé. La qualité et la structure d'une ontologie sont très importantes lorsqu'interviennent des questions d'interopérabilité et d'exactitude dans des contextes de raisonnement spatial. Généralement, «des ontologies riches en contenu et en structure favorisent une meilleure interopérabilité avec les bases de données géospatiales puisqu'elles offrent un plus grand éventail de concepts géospatiaux, de propriétés intrinsèques et de propriétés extrinsèquement» [Bro04]. Ceci s'explique par le fait que la richesse des ontologies influe sur la façon dont les données spatiales sont structurées. Ainsi, si les données sont bien structurées à la lumière d'ontologies riches en contenu et en structure, l'interopérabilité avec les données géospatiales sera meilleure et favorisera une bonne emprise sur les données.

Dans un deuxième temps, un traducteur est appliqué à l'ontologie géographique de même qu'aux données associées à celle-ci. Dans ce cas, le traducteur a été programmé en *Delphi*. Celui-ci lit dans des fichiers de bases de données les informations sur l'ontologie géographique et lit également les données. Il génère alors une représentation de l'ontologie géographique et des données en *PROLOG* incluant, entre autres, des faits, des règles de topologies, des règles de connexion et de superposition respectant les principes du *RCC* de Egenhofer. Ceux-ci forment une BGC. Dans un troisième temps, des règles de cohérence sont créées et des requêtes sont effectuées afin de vérifier la cohérence de la BGC et de déceler les incohérences.

Dans le but d'introduire le concept de base de géocroyances dans le cadre du présent travail, nous débutons par présenter les composantes de la définition de base de connaissances présentée par ces auteurs. Ceci s'effectue, entre autres, en présentant les différentes caractéristiques de la BNDT et de son ontologie relevées par les auteurs.

Définition d'une base de connaissance

Selon Mostafavi et Edwards, une base de connaissances constitue une représentation formelle d'une ontologie géographique. Pour les fins de leurs travaux, ils ont retenu le *PROLOG* en tant que support représentationnel. Nous nommons les bases de connaissances à contenu géographique «bases de géoconnaissances». Ainsi, la base de géoconnaissances créée dans [MEJ04a] est formée de faits et de règles exprimant des contraintes d'intégrité. Les règles servent à vérifier la cohérence des faits entre eux dans la base de géoconnaissances.

Des connaissances aux croyances

Les notions de connaissances ont soulevé bien des débats d'idées au cours des dernières années. En logique, les scientifiques s'entendent aujourd'hui habituellement pour affirmer que les connaissances sont considérées comme acquises, ce qui signifie qu'elles sont déjà prouvées et qu'elles ne peuvent faire l'objet de remises en question. Il en est tout autrement pour les croyances, lesquelles peuvent être remises en question et éventuellement être révisées afin de restaurer leur cohérence interne et mieux rendre compte de la réalité. Pour cette raison, dans le cadre de ce mémoire, nous développons le concept de base de géocroyances en géomatiques. Cependant, nous devons tenir compte de la littérature et des concepts qui ont précédé sur les connaissances et les bases de connaissances. Ainsi, nous nous servons de ces dernières notions pour définir celle de base de géocroyances qui rend bien compte de la réalité évolutive de l'information spatiale contenue la BNDT. Nous recourons également à des notions d'ingénierie des connaissances qui permettent de renforcer l'approche de structuration de la base de géocroyances que nous visons à définir au cours de ce chapitre.

L'ingénierie des connaissances appliquée à la définition d'une base de connaissances

Une base de connaissances est dans les faits, en ingénierie des connaissances d'après Schreiber (2000) [SAA+00], une composante intégrante d'un modèle de connaissances. Dans ce qui suit, en nous appuyant sur les travaux de Schreiber, nous proposons une définition de base de géocroyances qui permet de mieux tenir compte des besoins liés à l'usage de prédilection qu'avec une base de connaissances et de renforcer les ramifications avec les autres composantes d'un modèle de connaissances géographiques. Dans ce but, nous situons le concept de base de connaissances à l'intérieur d'un modèle de géoconnaissances.

3.2 Modèle de connaissances

Un modèle de connaissances est un modèle qui permet de bien cerner et modéliser les connaissances liées à un problème donné. Il nous permet également d'arriver à définir des bases de géocroyances (BGC) de manière rigoureuse dans le contexte de l'information spatiale en tenant compte des besoins spécifiques rattachés au problème traité. Selon Schreiber [SAA+00], un modèle de connaissances devrait comprendre au moins trois niveaux afin de garantir une bonne qualité de rigueur, de modélisation et de gestion des connaissances : la connaissance de la tâche, la connaissance d'inférence et la connaissance du domaine. Ces trois niveaux peuvent être décrits à l'aide de modèles de géoconnaissances.

3.2.1 La connaissance de la tâche

La connaissance de la tâche visée à l'intérieur d'un modèle de connaissance joue un rôle fondamental dans la façon de représenter les connaissances. Connaître quelles sont les différentes étapes impliquées dans une tâche aura un impact sur la façon de représenter les croyances intervenant dans la base de géocroyances que nous souhaitons construire. Dans le contexte des modèles de connaissances, Schreiber [SAA+00] suggère de décomposer en étapes les tâches de diagnostic comme une de celles qui nous intéresse, soit celle portant sur la cohérence de la BNDT. Ainsi, si une ou plusieurs tâches à réaliser sont simples, il sera possible d'utiliser un mode de représentation simple. Si les tâches s'avèrent plus complexes, il découle qu'un type de représentation d'un niveau de complexité plus élevé devra être envisagé.

Modèle de décomposition de la tâche de diagnostic

Le modèle de décomposition de tâche 3.4 présente les sous-tâches impliquées dans la tâche de diagnostic de la cohérence de la BNDT :

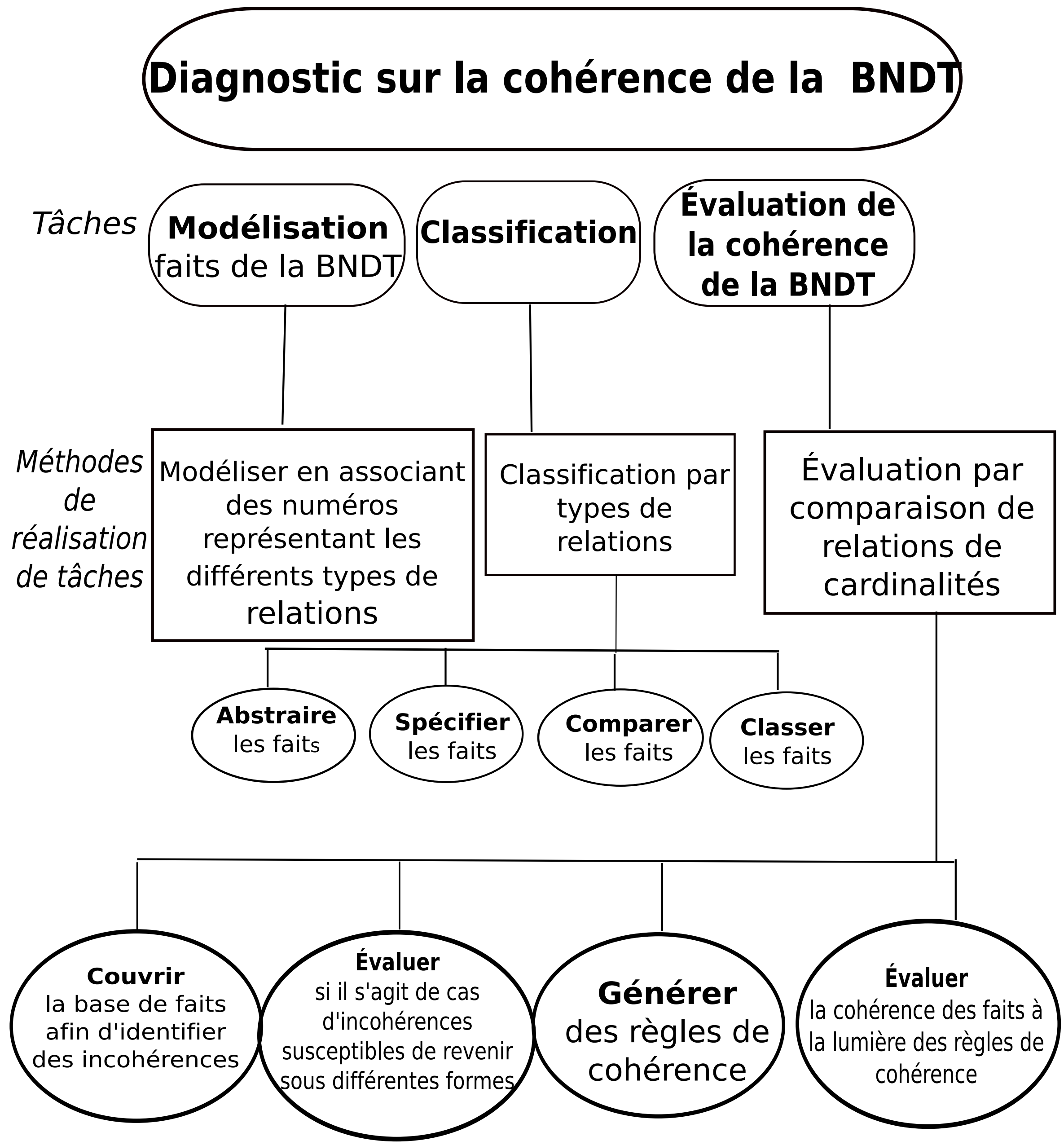

Fig. 3.2 – Modèle de décomposition de la tâche de diagnostic

3.2.2 La connaissance d'inférence

Ce type de connaissance permet de bien cerner le mécanisme de raisonnement logique qui conduit à tirer des conclusions dans une situation donnée. Il importe d'avoir une bonne emprise sur ce mécanisme pour effectuer des raisonnements rigoureux. Sa représentation sous forme de modèle conduit à bien identifier les étapes de raisonnement ainsi que les liens qui existent entre elles.

Modèle d'inférence

Ce modèle présente les différentes étapes de raisonnement qui conduisent à inférer des résultats. Le modèle d'inférence 3.3 est celui que que nous avons retenu afin d'inférer des résultats sur la cohérence de la BNDT :

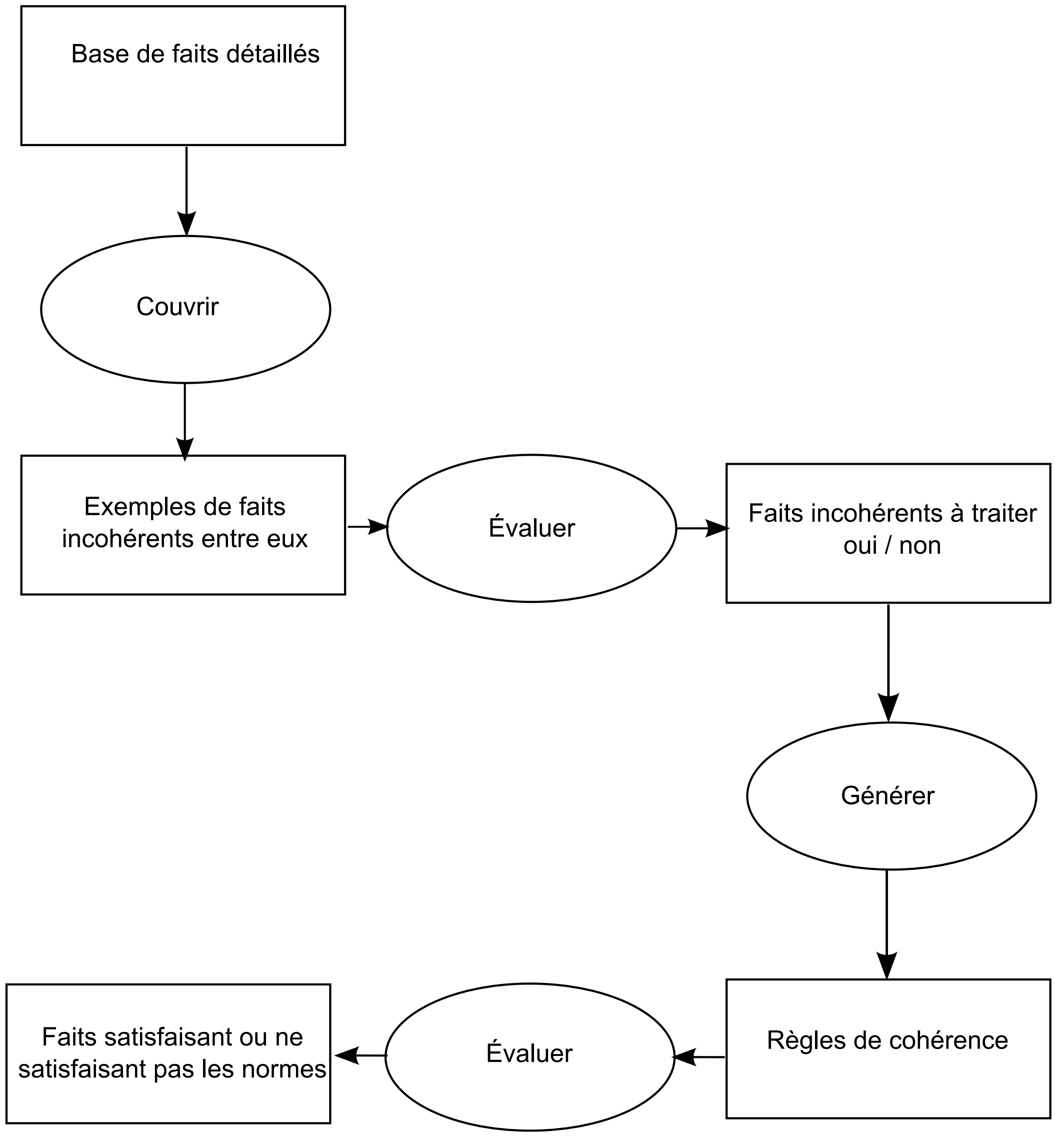

FIG. 3.3 – Modèle d'inférence sur la cohérence de la BNDT

La figure 3.3 présente les principales étapes à suivre lesquelles permettent à la fois de créer et de vérifier la cohérence d'une BGC. Dans un premier temps, une base de faits est créée à partir d'une ontologie comme celle de la BNDT. Ensuite, la base de faits est étudiée et couverte par un utilisateur. De cette manière en se référant à l'ontologie, il est en mesure d'identifier des types d'incohérences potentiels. S'il semble y avoir certains types d'incohérences, il devient nécessaire d'intervenir. Ceci conduit à devoir générer des règles de cohérences pour vérifier la cohérence. Ces règles qui traduisent les normes à respecter pour s'assurer de la cohérence de la BGC étudiée sont ensuites utilisées pour vérifier la cohérence.

3.2.3 La connaissance du domaine

Ce type de connaissance porte sur la nature des connaissances impliquées et traitées à l'intérieur du problème qui nous intéresse. La connaissance du domaine est formée de deux composantes principales, soient un schéma de domaine et une base de connaissances.

Schéma de domaine

Un schéma de domaine constitue une représentation du domaine à l'étude propre à un problème donné. Le schéma de domaine tel que proposé par Schreiber [SAA+00] comprend des concepts, qui correspondent aux différents thèmes liés au contexte d'un problème donné. Nous appliquons ce schéma de domaine à la BNDT et aux entités sous-jacentes aux différents thèmes qui sont impliquées dans des relations spatiales qui composent la BNDT. Ces relations spatiales constituent les variables de notre problème. Le schéma de domaine de notre problème contient principalement des relations de connexions, de partage, et d'adjacence et de superposition. Des règles de vérification de cohérence y sont également présentes. Un exemple de règle est détaillé dans le chapitre 4 pour la propriété de réflexivité de connexion.

La figure 3.4 présente le schéma de domaine que nous proposons pour la BNDT :

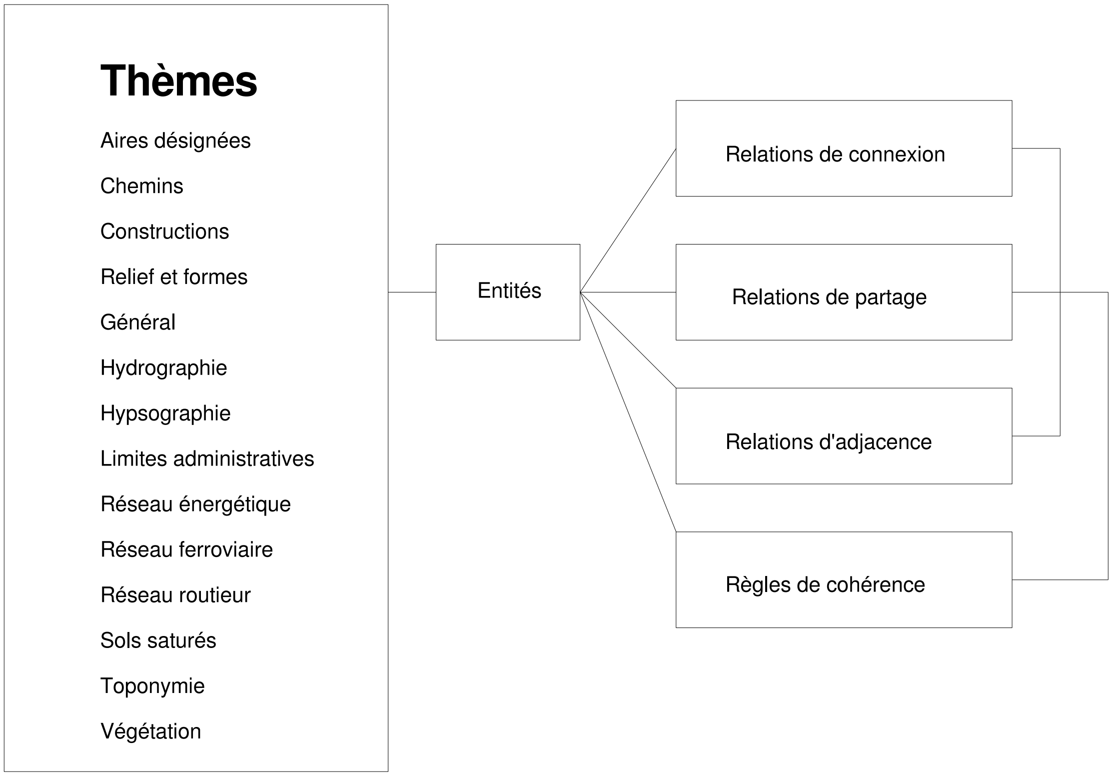

FIG. 3.4 – Schéma de domaine

Bases de géocroyances

Dans cette section, nous allons définir en détails ce qu'est une base de géocroyances. Pour ce faire, nous allons détaillé chacune des composantes d'une telle base. Les composantes d'une base de géocroyances sont essentiellement celles d'une base de connaissances mais dont le contenu est évolutif et de nature géographique. Les philosophes distinguent clairement les notions de «connaissances» et de «croyances». Les connaissances réfèrent à ce qui est acquis alors que les croyances réfèrent à ce qui est crû et, conséquemment, en constante évolution. Or, cela est très souvent le propre de l'information géographique d'évoluer et d'être régulièrement remise en question dans son exactitude et dans sa précision dans les systèmes d'information géographique (SIG). Ceci explique que nous soyons amenés à traiter de «bases de géocroyances» plutôt que de «bases de géoconnaissances» en géomatique.

Nous nous inspirons cependant d'une démarche et de modèles de connaissances proposés par Schreiber [SAA+00] pour parvenir à définir celui de bases de géocroyances.

Une base de connaissances repose sur la définition d'un schéma de domaine. Selon Schreiber, "un schéma de domaine est une description schématique d'une connaissance et d'une information propres à un domaine spécifique à travers un certain nombre de définitions de types" [SAA+00]. Selon le même auteur, un schéma de domaine peut contenir des généralisations de structures de domaines. Également, un schéma d'un point de vue de génie logiciel ressemble à un modèle de données ou encore à un modèle-objet.

En fait, une base de connaissances est un ensemble d'instances de différents types de connaissances présentes dans un schéma de domaine telles que des concepts, des relations et des règles, tel que présenté par Schreiber [SAA+00].

Nous définissons donc une base de géocroyances comme un ensemble d'instances de différents types de croyances de nature géographique présentes dans un schéma de domaine telles que des concepts, des relations et des règles, tel que présenté par Schreiber [SAA+00]. Ainsi, une base de croyances permet de tenir compte du caractère évolutif des croyances que nous avons sur l'information spatial en des lieux donnés contrairement à une base de connaissances.

La description fournie par un schéma de domaine couvre le «monde des possibles», lequel illustre l'ensemble des éventualités envisageables à l'intérieur d'un monde statique que nous considérons fini. Les types spécifiés à l'intérieur du schéma de domaine peuvent être représentés par des variables correspondant à diverses entités telles que des constructions, des limites administratives et des réseaux routiers. Le schéma de domaine s'étend également aux relations qui existent entre les entités comprises dans les relations spatiales. Chacune des instances « possibles » de ces variables de la BNDT sont organisées en intervalles d'appartenance qui sont réservés pour chacun des différents thèmes. Ces instances sont les éléments constitutifs de notre base de croyances. Vu leur caractère géographique et évolutif, nous les appelons géocroyances. La base de géocroyances comprend les connaissances sur le domaine de même que les contraintes d'intégrité sur le domaine exprimées sous forme clausale.

3.3 Représentation des croyances dans une base de géocroyances

Trouver un mode de représentation des croyances qui soit adéquat présente souvent des difficultés, puisqu'un grand nombre de choix de représentation des croyances doit être réalisé en anticipant les modifications que peuvent subir les croyances au fil du temps et celles qui viennent s'ajouter. Pour notre problème, nous avons choisi de recourir à la logique afin de représenter les croyances qui nous intéressent. Ce choix est motivé, entre autres, par le type d'analyse que nous effectuons, soit des analyses de cohérence logique. La logique se prête bien pour les analyses spatiales qualitatives. Nous reconnaissons cependant, que pour des analyses spatiales quantitatives les possibilités sont plus limitées par le formalisme qui n'est pas destiné en premier lieu au calcul numérique. Ainsi, la logique constitue un support représentationnel qui sera très utile pour effectuer des raisonnements spatiaux qualitatifs sur des géocroyances, un aspect que nous traiterons de façon plus détaillée dans le quatrième chapitre de ce mémoire.

Pour les fins de notre travail, la logique du premier ordre suffit pour représenter nos croyances afin de répondre à nos besoins. Nous parvenons à représenter les croyances géographiques du problème de la BNDT sous forme de faits et de règles. Un exemple détaillé de base de géocroyances est donné à la section 4.7 de ce mémoire dans la section «Révision des géocroyances». Dans ce qui suit, nous présentons l'ontologie de la BNDT et la terminologie dont nous nous dotons pour former une base de géocroyances qui soit adaptée à notre problème.

3.4 Description de l'ontologie de la BNDT et caractéristiques d'une base de géocroyances

Dans cette section, nous décrivons les origines de la BNDT et nous en effectuons une description générale en présentant ses différentes composantes et caractéristiques. Nous établissons le lien entre ses composantes et son ontologie, ses propriétés ainsi que différentes représentations possibles qui s'y rattachent.

Origines de la BNDT

L'organisme Géomatique Canada a amorcé en 1990 la création de la BNDT. Les objectifs premiers visés par la BNDT étaient de répondre aux besoins du gouvernement canadien en termes de production cartographique et d'être en mesure d'offrir aux utilisateurs de technologies géomatiques, tels que les SIG, des données topographiques numériques.

En 1994, des améliorations importantes ont été apportées à la BNDT dans le but de mieux répondre aux besoins des utilisateurs suite à une rencontre avec ces derniers. Le concept de la BNDT a alors considérablement évolué sur le plan des normes, des spécifications et des données présentes à l'intérieur de celle-ci.

Dans le cadre de ce mémoire, nous nous appuyons sur l'édition 3.1 des *Normes et spécifications de la Base nationale de données topographiques*. Ce document facilite la compréhension de l'organisation des composantes de la BNDT. Ces composantes sont les jeux de données, les thèmes, les entités, les occurrences d'entité et de relations spatiales.

Description générale de la BNDT

La BNDT contient un grand éventail de jeux de données topographiques provenant du Système national de référence cartographique (SNRC). Elle est constituée d'informations propres à 112 entités lesquelles sont regroupées suivant 14 thèmes différents.

Les 14 thèmes de la BNDT sont les suivants :

- Aires désignées
- Chemins
- Constructions
- Relief et formes

- Général
- Hydrographie
- Hypsographie
- Limites administratives
- Réseau énergétique
- Réseau ferroviaire
- Réseau routier
- Sols saturés
- Toponymie
- Végétation

Pour les fins de gestion de la BNDT, le territoire canadien fait l'objet d'une subdivision territoriale qui épouse le découpage régulier du SNRC. Ce découpage est effectué à l'échelle 1 :50 000 et respecte les normes établies par le Système de référence géodésique nord américain datant de 1983 communément appelé NAD83.

Voici un schéma présentant la structure générale de la BNDT.

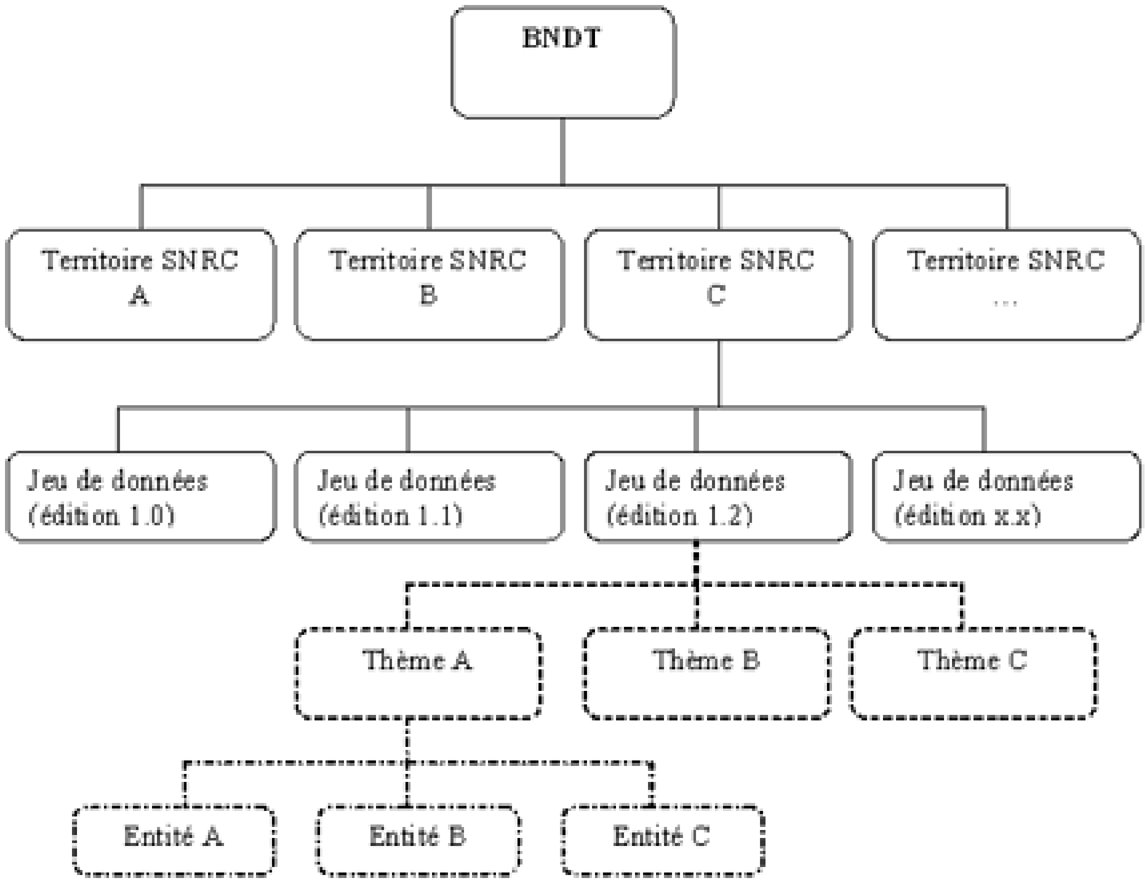

Fig. 3.5 – Structure générale de la BNDT

Sur la figure 3.5, nous voyons que la BNDT a été élaborée suivant plusieurs étapes.

La figure 3.6 permet de voir l'échelle à laquelle est disponible l'information couvrant différents secteurs de la BNDT sur le territoire pancanadien :

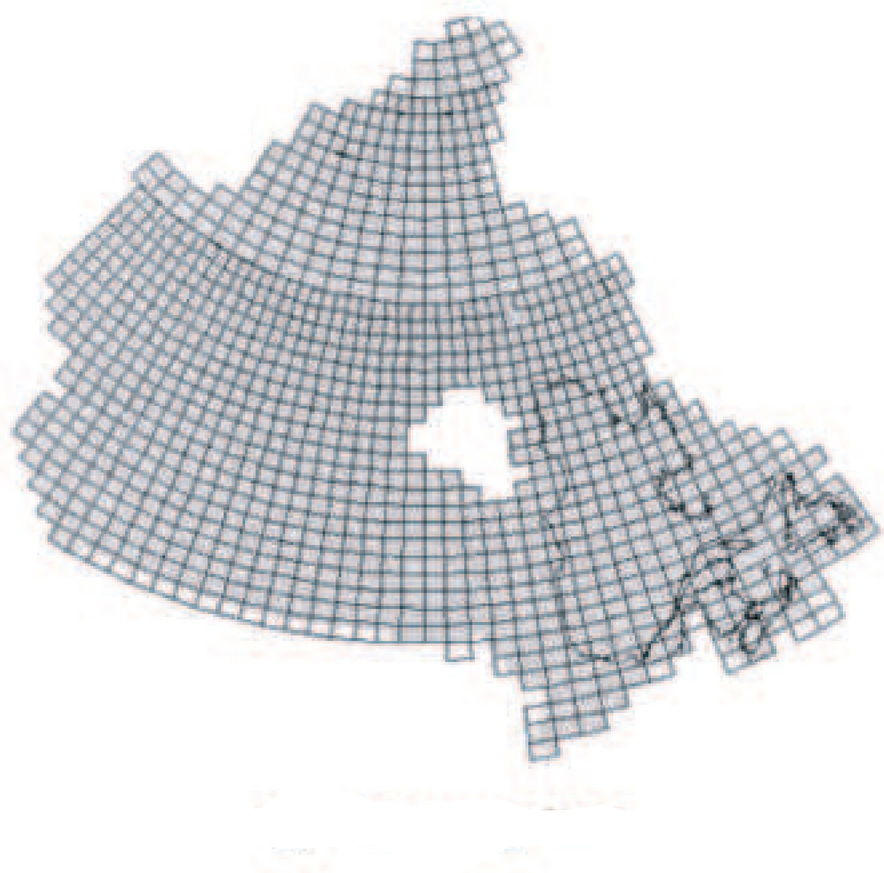

FIG. 3.6 – Découpage SNRC de la BNDT

Dans la figure 3.6, il est possible de visualiser les secteurs couverts par la cartographie à l'échelle 1 :50 000 qui sont complétés par la cartographie 1 :250 000 lorsque l'information à une telle précision n'est pas rendue disponible par le CITS [CIT]. Particulièrement, dans le Nord Canadien, il est visible que la cartographie 1 :50 000 n'est pas disponible et que cette région est représentée par des «carrés» couvrant de plus grands secteurs, soit à l'échelle 1 :250 000.

Étapes de création d'une base de connaissances exprimée en *PROLOG*

Dans la figure 3.7, issue de [MEJ04a], il est possible de voir un exemple d'étapes conduisant à la création d'une base de connaissances et à la vérification de connaissances réalisé à partir de la BNDT par Mostafavi [MEJ04a].

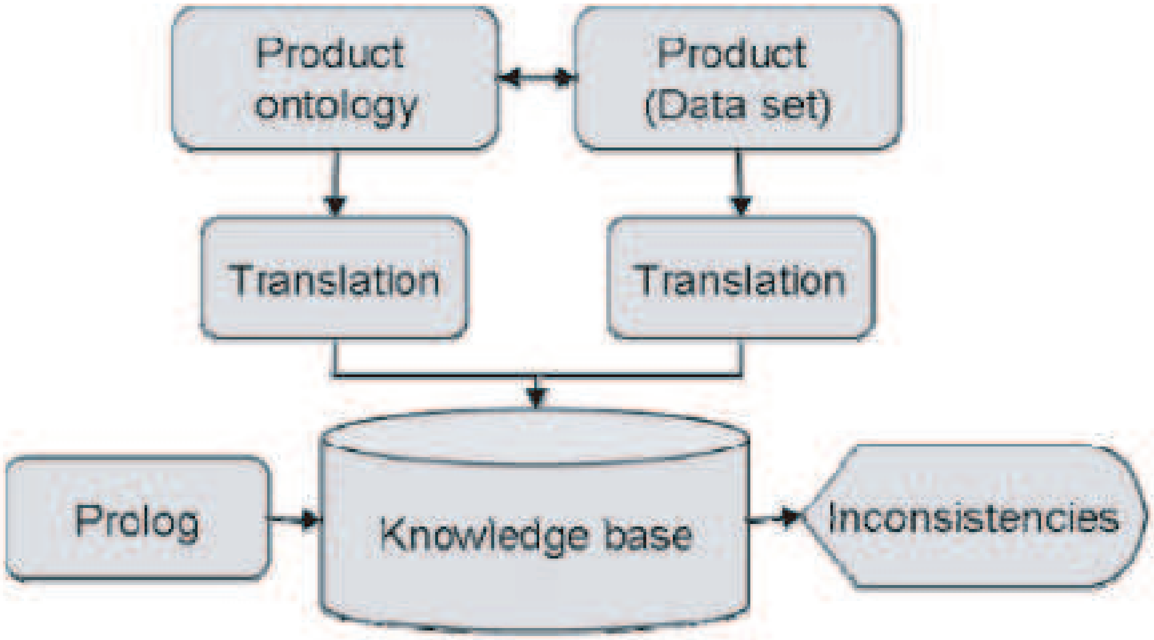

FIG. 3.7 – Étapes de conception d'une base de connaissances et vérification de la cohérence ontologique

3.4.1 Terminologie de la BNDT

Afin de désigner distinctivement les diverses combinaisons de valeurs d'*attributs* d'objets spatiaux impliquées dans différents types de relations (ex : connexion, partages, superposition), un *code* unique est assigné à des combinaisons de valeurs d'attributs d'une entité pour une représentation géométrique donnée. Par *entité*, nous entendons le nom de l'entité tel qu'il apparaît dans le *Dictionnaire des données de la BNDT*. Par *combinaison*, nous faisons allusion aux combinaisons autorisées de valeurs d'attributs au sein de la BNDT. Lorsque nous traitons des *cardinalités de relation*, nous faisons référence au nombre minimum et maximum de fois qu'une occurrence d'entité peut entrer en relation, soit de connexion, de partage ou de superposition, avec une autre occurrence d'entité.

3.4.2 Variables

Les variables de notre problème sont : les entités des éléments impliqués dans des relations spatiales caractérisant notre problème. Pour les relations de connexion, qui nous intéressent principalement, nous avons les variables U, V, X et Y. Les variables U et V servent à caractériser les types de connexion alors que les variables X et Y servent à désigner les numéros d'entités impliqués dans les relations de connexion. Une relation de connexion entre deux entités aura la forme $c(U, V, X, Y)$.

3.4.3 Domaine

Le domaine est constitué de l'étendue des numéros des entités des valeurs de la BNDT, soient les valeurs 0 à 6000 pour les variables X et Y. Les variables U et V peuvent prendre les valeurs $\{0, 1, 2\}$.

La cardinalité des relations à l'intérieur du domaine est définie par le nombre d'instances impliquées dans la relation (0,0 : aucune ; 0,1 : possibilité d'au plus une instance ; 0,2 : possibilité d'au plus deux instances ; 1,2 : obligatoirement une instance et possiblement deux : 2,2 : obligatoirement deux et seulement deux instances.

3.4.4 Régles et contraintes d'intégrité

Nous définissons D en tant que domaine fini pour lequel S_D désigne l'ensemble de toutes les possibilités combinatoires de codes exprimées sous forme de clauses et S_C en tant qu'ensemble de clauses représentant les relations entre les variables traduisant des contraintes d'intégrités. Ces relations peuvent être, entre autres, des relations de symétrie. La propriété de symétrie se traduit par : A r B si et seulement si B r A. Cet ensemble de clauses contient les règles de cohérence. S_O constitue l'ensemble de clauses représentant les géocroyances constituant des faits. S_O correspond plus précisément dans notre contexte d'application à l'ontologie de la BNDT initiale représentée par Mostafavi et al [MEJ04a] en *PROLOG* que nous traduisons en *ASP*).

Unité de mesure

L'unité de mesure retenue pour les unités de positionnement des données spatiales en coordonnées (x, y) est le mètre en valeur entière.

Précision géométrique

La précision des données d'une occurrence d'entité est représentée par la distance entre la position de sa représentation géométrique et sa position mesurée sur le terrain en lien avec le canevas géodésique.

Résolution spatiale

Lorsque nous faisons allusion à la résolution, nous référons à la taille des éléments présents à l'intérieur de la BNDT. Celle-ci est définie à l'aide d'un ensemble de "dimensions garanties" qui caractérisent les dimensions des éléments topographiques que la BNDT couvre.

LA BNDT comporte deux classes de résolution, les territoires de fortes densités correspondant au milieu urbain ou rural sont dans la majorité des cas disponibles au 1 :50 000. La cartographie de milieu est pour sa part généralement disponible au 1 :250 000.

Structure spatiale

Selon les responsables de la BNDT, celle-ci est exempte de toute incohérence de nature spatiale. Nous verrons cependant dans les faits qu'un certain nombre d'incohérences se trouvent au sein de la BNDT. Ces incohérences se retrouvent, entre autres, au niveau des relations de connexion. Les données sont exemptes de toute autre incohérence de nature spatiale, telles que des dépassements, des espacements ou des erreurs de fermeture de surface. Des relations de connexion et des relations de superposition exprimées sous formes de règles logiques servent à assurer l'intégrité de la structure spatiale.

Qualité des données

Le Centre d'information topographique (CIT) constitue l'organisme qui veille à la qualité des données de la BNDT. Le CIT vérifie que le contenu topographique de la BNDT s'accorde bien avec le contenu topographique de la source de données utilisée suivant les Normes et spécifications de la BNDT.

Trois types d'erreurs sont généralement rencontrés dans les données provenant de la BNDT :

1. Erreurs de classification : Erreurs survenant lors d'une mauvaise attribution de code.

2. Erreurs d'omission : Erreurs survenant lorsqu'un objet est présent dans la source de données mais est absent du jeu de données.
3. Erreurs de commission : Erreurs survenant lorsque qu'un objet est présent à l'intérieur d'un jeu de données mais qu'il ne figure pas à l'intérieur d'une source de données.

Les responsables de la BNDT associent généralement un taux d'erreurs maximal de cinq pour cent à ces différents types d'erreurs à l'intérieur d'un échantillon représentatif.

Toutefois, dans le contexte actuel, nous nous intéressons plus spécialement aux erreurs à l'intérieur de l'ontologie même de la BNDT et non à celles des sources de données de la BNDT. Ces erreurs sont :

1. Erreurs de classification : Erreurs survenant lors d'attribution de codes inexistants, c'est-à-dire lorsque les codes n'appartiennent pas au domaine du problème et ne font donc pas partie du monde des possibles.
2. Erreurs d'omission : Erreurs survenant lorsqu'une relation spatiale n'est pas définie symétriquement.

Ces erreurs pourront être décelées à l'aide de règles de cohérence qui seront détaillées dans le chapitre 4. Ces règles seront introduites à l'intérieur d'une BGC.

3.5 Bilan de chapitre

Au cours de ce chapitre, nous avons vu comment les bases de connaissances et les bases de géocroyances se définissent habituellement. Nous avons exploré l'avenue de l'ingénierie des connaissances pour enrichir la notion de base de connaissance en la situant à l'intérieur d'un modèle de connaissances. Travaillant dans un contexte où l'information géographique évolue régulièrement nous avons été amené à proposer l'introduction de la notion de base de géocroyances qui constitue un outil de grande utilité pour le raisonnement spatial que nous abordons de façon détaillée dans le chapitre suivant.

Chapitre 4

Le raisonnement spatial à l'intérieur d'une BGC

Dans ce chapitre, nous appliquons le raisonnement logique à un exemple de BGC constituant une représentation d'un contexte spatial caractéristique tiré de la BNDT. Les BGC telles que celles que nous concevons pour résoudre notre problème, soit celui de l'étude de la cohérence d'une BGC, se prêtent bien à l'utilisation de moteurs d'inférence de raisonnement logique tels que *PROLOG* et *Smodels*. Nous débutons en expliquant comment le raisonnement spatial s'applique au contexte de la BNDT et nous décrivons son fonctionnement. Nous étudions l'application du raisonnement monotone au contexte de la BNDT et les limites qu'il pose sur le plan de son fonctionnement. À la lumière de ces constats, nous présentons de nouvelles possibilités qu'offre le raisonnement non monotone. Nous expliquons, par la suite, comment nous procédons à une vérification de la cohérence d'un échantillon représentatif de la BNDT que nous avons sélectionné. Nous décelons des incohérences au sein d'une BGC, ce qui nous amène à nous intéresser au problème de révision des BGC pour rétablir leur cohérence. Nous décrivons alors le fonctionnement de la méthode de révision des croyances des r-ensembles développée par Odile Papini [Pap92] et appliquée aux connaissances géographiques non stratifiée dans [Wur00]. Réalisant que cette approche cadre bien avec les besoins liés à notre problème, nous regardons comment cette méthode s'applique à une BGC de la BNDT.

4.1 Raisonnement spatial dans la BNDT

Le raisonnement spatial constitue un processus déductif permettant de tirer des interprétations sur les relations qui lient différents objets de nature spatiale dans un environnement physique. Ces interprétations effectuées à partir d'informations spatiales permettent de tirer des conclusions sur des propriétés comme, par exemple, la position et la distance qui lient des objets spatiaux. De telles conclusions peuvent également être tirées sur des relations de voisinage, des relations telles que «faire partie de», «être composé de» et des notions telles que la granularité de la référence spatiale qui sont décrites par Bittner et Edwards dans [BE01]. L'utilisation de ce type de raisonnement nécessite de bien identifier les variables physiques des problèmes que nous visons à résoudre dans un contexte donné. Il implique une cueillette d'informations spatiales qui serviront à définir le domaine des variables du problème que nous cherchons à résoudre.

Dans ce travail, nous nous concentrons sur la dimension qualitative du raisonnement spatial. En raisonnant qualitativement sur une BGC, nous visons à restaurer sa cohérence afin d'être en mesure d'en améliorer l'intégrité. Ceci permet ensuite d'offrir une meilleure garantie sur la qualité des analyses spatiales qualitatives et quantitatives effectuées ultérieurement.

Afin de pouvoir raisonner spatialement sur la BNDT, nous choisissons de représenter notre problème sous forme d'expressions logiques. Le recours à une approche logique présente un intérêt particulier lorsque nous travaillons au niveau ontologique sur des problèmes de raisonnement qualitatif de nature spatiale. Une telle approche permet dans certaines situations, que nous exposerons, de déceler rapidement les incohérences qui existent au niveau ontologique. Nous verrons également l'intérêt qu'offre une représentation logique pour la révision de géocroyances.

Pour nous convaincre de l'efficacité du recours à des approches logiques, nous représentons notre problème suivant deux langages de programmation logiques différents. Dans un premier temps, nous représentons notre problème en *PROLOG* et dans un second temps, nous le représentons en *ASP*. De cette manière, il devient possible d'effectuer des opérations de raisonnement logique en utilisant des moteurs de raisonnement propres à ces deux langages, soient respectivement un moteur d'inférence *PROLOG* et le moteur d'inférence *Smodels*. En l'occurrence, nous partons de la représentation des connaissances géographiques de la BNDT proposée par Mostafavi (2004) en *PROLOG* que nous simplifions afin de tenir compte des contraintes de représentation propres à l'*ASP* et de proposer des représentations équivalentes en *PROLOG* et en *ASP* pour les fins de nos comparaisons.

4.2 Raisonnement monotone dans la BNDT

Le raisonnement monotone est un type de raisonnement qui s'applique bien à de nombreuses situations et qui reste relativement simple, en ce sens que lorsqu'une conclusion a été tirée une fois lors de l'application de ce type de raisonnement, cette conclusion n'est plus jamais remise en cause, et ce, même lors de l'ajout de nouvelles informations. Le raisonnement monotone est adéquat lorsque l'ajout de nouvelles informations ou lorsque la fusion d'ontologies n'entraînent pas l'apparition d'incohérences.

Exemple :

Soit le programme Δ, dans lequel r désigne une route, b un bâtiment et c une relation de connexion entre deux entités géographiques :

$c(r,b)$.
$c(b,r)$.
$coherent(r,b)$:- $c(r,b)$, $c(b,r)$.

Une nouvelle route croisant un chemin de fer est construite. Ceci donne naissance à une nouvelle relation de connexion entre cette nouvelle route et un chemin de fer. Nous ajoutons au programme le fait $c(r,ch)$ qui désigne une relation de connexion entre cette route et un chemin de fer.

Le programme Δ devient :

$c(r,ch)$.
$c(r,b)$.
$c(b,r)$.
$coherent(r,b)$:- $c(r,b)$, $c(b,r)$.

En ajoutant ce fait, nous avons effectué une mise à jour des géocroyances du programme Δ. Cet ajout d'information n'a pas affecté la cohérence du programme et a entraîné son expansion.

Une autre situation qui revient est le cas du traitement d'exceptions simples qui contiennent des règles contenant un seul prédicat dans la queue de la règle. Voici un exemple de programme Δ contenant ce type de règle :

$\Delta =$

$c(12, 18)$.
$coherent(X, Y)$:- $c(X, Y)$, $NOT\ c(40, 50)$.

La règle présente dans ce programme contient uniquement le prédicat c dans sa queue. Ce type de programme contenant une règle visant à déceler le cas d'exception dans lequel $c(40, 50)$ est présent, se traite très bien autant en $PROLOG$ qu'en ASP. En ASP, nous obtiendrons que "$coherent(12, 18)$" fait partie d'un modèle stable alors qu'en $PROLOG$, nous aurons que la requête "$coherent(X, Y)$." est vérifiée.

Toutefois, des situations aussi simples sont rares dans le contexte spatial et lorsque de l'information incomplète doit être utilisée et que des incohérences surviennent, il devient intéressant, voire nécessaire, de se tourner vers un autre type de raisonnement qu'est celui du raisonnement non monotone pour rétablir la cohérence.

Dans l'exemple qui suit prédicat s désigne une relation de superposition entre deux entités géographiques. Dans le cas où nous aurions un programme contenant un prédicat indéfini dans la partie NOT tel que le suivant :

$\Delta =$

$c(12, 18)$.
$coherent(X, Y)$:- $c(X, Y)$, $NOT\ s(40, 50)$.

Nous n'arriverions plus à le traiter en $PROLOG$. La requête "$coherent(X, Y)$." donnerait une erreur d'indéfinition du prédicat $s(40, 50)$. Un tel programme en $PROLOG$ suit globalement le fonctionnement de la logique classique. Le même programme abordé en ASP avec le moteur de raisonnement $SMODELS$ serait fonctionnel et la règle qu'il contient serait valide. Ainsi, $coherent(12, 18)$ serait vérifié et appartiendrait à un modèle stable du programme Δ. Le programme, lorsque traité en ASP, suit essentiellement le fonctionnement de la logique des défauts, une logique non monotone. Ainsi, l'atome $s(40, 50)$ est considéré non vérifié, jusqu'à preuve du contraire.

4.3 Raisonnement non monotone dans la BNDT

Le raisonnement non monotone offre l'avantage de rendre possible de traiter l'information incomplète dans des contextes spatiaux d'application comme celui de la BNDT.

L'*Answer Set Programming* (ASP), un langage de programmation logique permettant de mettre en oeuvre le raisonnement non monotone, rend réalisable le traitement d'information incomplète de la BNDT. Ceci devient concrétisable avec l'utilisation de *Smodels*, un moteur de raisonnement de type "Solver" pour l'*ASP* qui permet de recréer le raisonnement non monotone à l'intérieur d'une BGC. L'utilisation de *Smodels* offre des possibilités représentationelles supplémentaires, dont celle de représenter de l'information incomplète plus facilement qu'en *PROLOG* et de permettre l'implantation de stratégies de révision sans nécessiter trop de lignes de programmation.

Un cas intéressant d'application est celui des relations de connexion qui coïncident avec des relations de superposition pour un même objet donné. Un exemple typique tiré de la BNDT est celui d'une route dont la relation de connexion avec un bâtiment est permise et dont la relation de superposition entre ces deux mêmes objets est interdite. Or, dans la réalité il existe, entre autres, un cas d'exception qui est celui d'un hôpital pour lequel ces deux types de relation simultanés sont permis. Ainsi, les contraintes d'intégrité liées à cette réalité qui tient compte de ce cas d'exception peuvent s'exprimer à l'aide d'une expression logique permettant de détecter des incohérences telles que la suivante :

(1) $coherent(r, b)$:- $c(r, b)$, $s(r, b)$ not $c(r, h)$, $s(r, h)$.

La règle ci-dessus se lit : S'il existe une relation de connexion entre une route r et un bâtiment b et une relation de superposition entre une route r et un bâtiment b et qu'on ne peut pas prouver qu'il n'y a pas de relation de connexion et de superposition entre cette même route r et un bâtiment qui est un hôpital h, alors les relations qui lient la route r et le bâtiment b sont cohérentes.

L'avantage que comporte l'utilisation de *Smodels* dans le cas d'une telle application est qu'il est possible d'utiliser un programme comportant de l'information incomplète tel que des prédicats manquants représentant des informations spatiales. Dans un programme logique dans lequel la règle (1) figurerait, le programme demeurerait valide même si le prédicat de la partie NOT, soit s de l'atome $s(r, h)$ était indéfini. Ainsi, si dans une BGC nous ignorons s'il existe effectivement un hôpital, le prédicat peut demeurer indéfini et la règle (1) reste fonctionnelle et valide dans un tel cas d'indétermination. En *PROLOG*, dans sa forme originale exempte d'extensions, le programme deviendrait non fonctionnel et ne serait pas utilisable.

Voici une image présentant ce cas d'incohérence se retrouvant dans la BNDT :

Fig. 4.1 – Exemples d'incohérence au sein de la BNDT : Ambulance sur une route sous un hôpital

Un exemple semblable est celui du cas dans lequel nous savons que la superposition entre une rivière et un lac est interdite mais que nous ignorons si dans la BGC que nous étudions ce cas se présente. Cette situation est difficilement traitable en *PROLOG* car le moteur de raisonnement refusera de fonctionner si un prédicat dans une règle est indéfini tel qu'il est résumé dans le schéma suivant :

Le schéma ci-dessus montre une limite importante que pose l'utilisation du *PROLOG* pour la représentation et le traitement de l'information incomplète. Nous voyons que dans un tel cas, il est souhaitable de se tourner vers d'autres langages comme l'*ASP* pour la représentation des croyances sinon il est nécessaire de s'en remettre à l'usage d'extensions complexes de *PROLOG* qui accroissent considérablement la difficulté de la tâche comme dans [MHV05].

Des implantations comme *Smodels* se montrent également très utiles pour des situations dans lesquelles des conclusions tirées antérieurement deviennent inexactes à la suite de l'ajout de nouvelles informations. Le moteur de raisonnement *Smodels* autorise le retrait de conclusions précédemment établies lorsque des prédicats indéfinis présents dans la partie *NOT* deviennent définis lors de l'ajout de propositions supplémentaires. Ainsi, une conclusion initialement vraisemblable peut ensuite être invalidée lors de l'ajout de nouvelles propositions et conduire à de nouvelles conclusions. Dans de nombreuses situations spatiales, l'outil de raisonnement non monotone *Smodels* est très pratique car il permet de réviser les croyances que nous avons sur un domaine et ainsi de restaurer la cohérence à la suite de la mise en commun de bases de géocroyances. Dans la figure 4.2, nous voyons une évolution des conclusions en ASP en comparaison avec le schéma précédent :

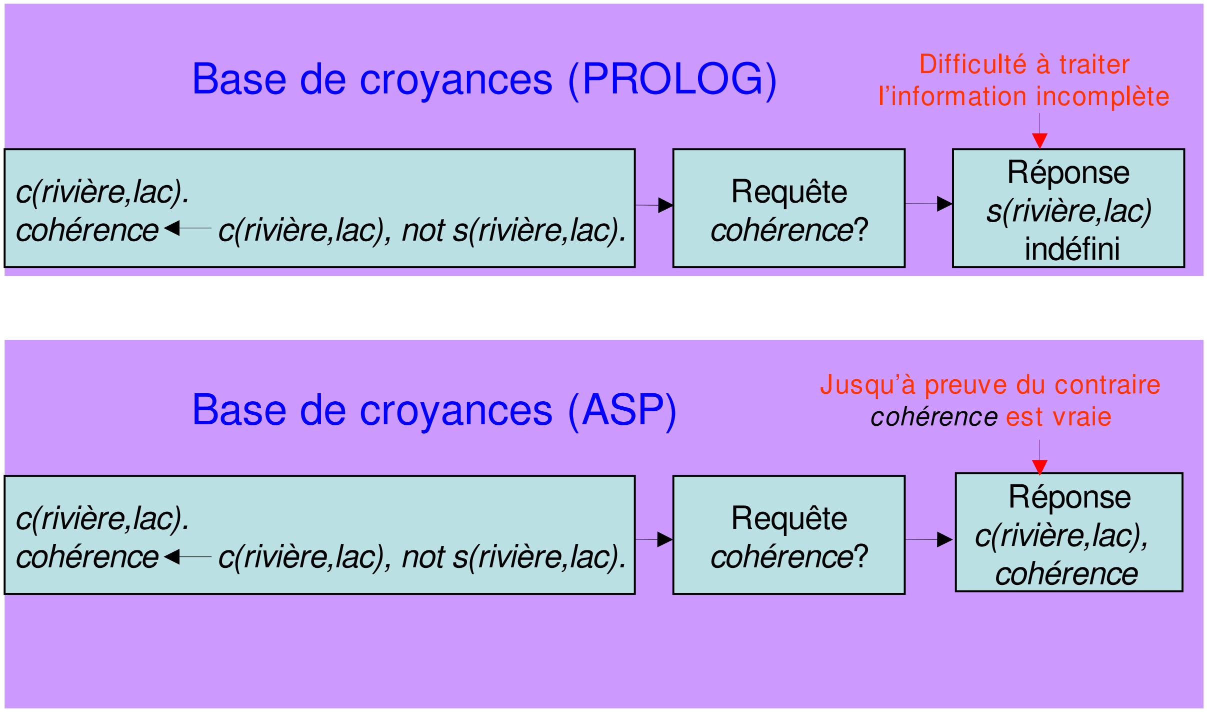

FIG. 4.2 – Ajout d'une information au sein de la BNDT : *PROLOG* vs *ASP*

Dans la figure 4.2, nous voyons au niveau de la flèche C qu'il y a eu retrait de la conclusion *cohérence*. Nous notons également au niveau de la flèche B qu'il y a eu l'ajout de la conclusion $s(riviere, lac)$. Nous observons également que le programme de *PROLOG* devient fonctionnel car l'ajout de l'information $s(riviere, lac)$ "complète" le programme qui était initialement incomplet.

En *ASP*, dans le cas où un programme logique devient incohérent et que le retrait de conclusions effectué automatiquement lors de l'ajout de nouvelles informations ne suffit pas à maintenir la cohérence du programme, il devient nécessaire d'identifier des faits (atomes) du programme à retirer. Cette dernière opération se nomme "révision". Une approche intéressante pour réviser les géocroyances consiste à recourir à une version adaptée de *Smodels*. L'opération de révision et une version adaptée du moteur de raisonnement *Smodels* [Pa03] seront présentés au cours de ce chapitre. La révision est une opération qui conduit à rétablir la cohérence dans une BGC en éliminant des faits de manière stratégique de façon à rétablir la cohérence de la BGC visée. Avant de nous intéresser à l'opération de révision et à la version adaptée du moteur de raisonnement *Smodels*, nous abordons la vérification de la cohérence ontologique qui nous permet d'évaluer l'ampleur du nombre d'incohérences résidant dans une BGC.

4.4 Traduction d'un programme logique

La traduction d'un programme logique exprimé en *PROLOG* en *ASP* est un processus nécessaire car la syntaxe des deux langages diffère à plusieurs niveaux. En fait, la syntaxe de *PROLOG* est généralement plus permissive que celle de l'*ASP*. Une des principales raisons qui impose une traduction est l'absence du concept de «Liste» en *ASP*. Ceci constitue un handicap important. Le principal rôle du présent traducteur est donc de décomposer les listes présentes dans un programme logique exprimé en *PROLOG* pour qu'il puisse ensuite être utilisé par l'outil *Smodels*. En résumé, le traducteur *PrologASPtr*, tel que conçu, lit un fichier contenant un échantillon de la BNDT exprimé en *PROLOG*. Il détecte la syntaxe propre au *PROLOG* et traduit le programme logique en *ASP*.

4.5 Vérification de la cohérence ontologique

La vérification de la cohérence ontologique se révèle un processus essentiel afin de déterminer s'il existe des incohérences à l'intérieur d'une base de géocroyances surtout à la suite d'une opération de fusion de BGC. La présence d'incohérences devient particulièrement problématique lors d'opérations d'analyses spatiales, car les résultats de ces dernières risquent de devenir obsolètes, ce qui explique l'intérêt pour ce problème. Cela est notamment vrai dans le cas de la BNDT, par exemple. Il s'ensuit que la tâche de vérification de cohérence se montre très utile pour déceler les incohérences qui résident à l'intérieur de la BNDT. Toutefois, une telle tâche ne permet pas de bien cerner la portée des incohérences prises individuellement.

Dans le but d'effectuer une vérification de cohérence, nous avons représenté en *PROLOG* et en *ASP* les croyances géographiques ontologiques que nous possédons sur un échantillon de la BNDT que nous avons choisi. De telles représentations nous ont ensuite permis d'accomplir une vérification de cohérence ontologique suivant trois étapes que sont :

1. La représentation du problème dans des formalismes appropriés, soit le *PROLOG* et l'*ASP* dans notre cas.
2. Le choix de moteurs de raisonnement adéquats pour ces formalismes logiques.
3. Les requêtes sur la cohérence ontologique.

La figure 4.3 résume ces étapes :

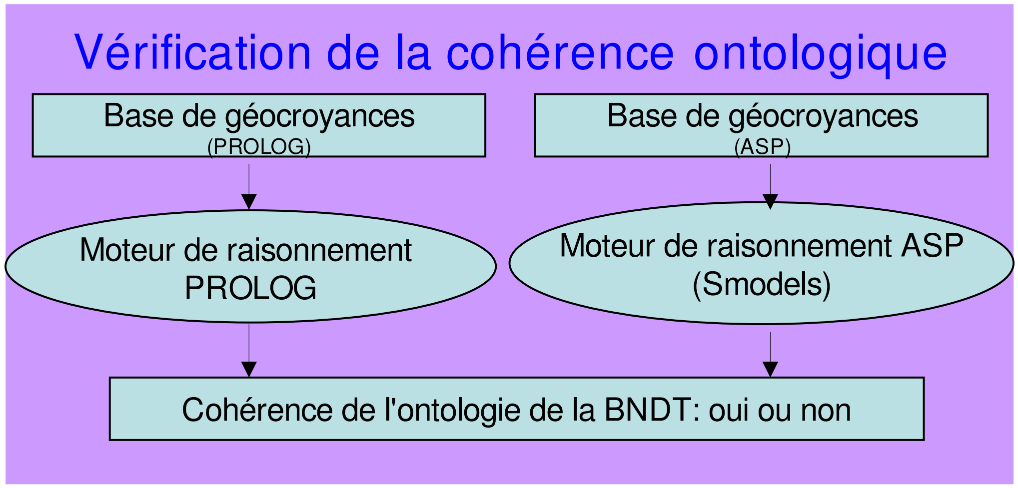

Fig. 4.3 – Étapes de la vérification de cohérence ontologique

Au cours de notre phase d'expérimentation, nous avons vérifié la cohérence de BGC de différentes tailles. Nous sommes parvenus à traiter en ASP avec Smodels 2.28 une BGC de 341 948 atomes correspondant à un ensemble de 174 654 faits représentant des relations de connexions entre des objets géographiques appartenant à la BNDT. Ainsi, lors de ce traitement, nous avons vérifié la propriété de symétrie de connexion pour l'ensemble de ces objets géographiques. Par symétrie de connexion nous entendons que si un objet géographique A est connecté à un object géographique B, alors le même objet B doit également être connecté à l'objet A. La propriété de symétrie de connexion s'exprime comme il suit : A r B si et seulement si B r A. Le nombre de 341948 a été le nombre critique d'atomes que nous sommes arrivés à traiter avant que n'apparaisse le message d'erreur "Error in input" lors de l'ajout d'une expression logique supplémentaire. Ce calcul s'est effectué en 5.830 secondes. Ceci nous laisse croire qu'il serait intéressant d'étudier la possibilité d'ajuster certains paramètres de *Smodels* afin de vérifier s'il est concevable de traiter des programmes de plus grandes tailles. En effet, le problème initial que nous visions à traiter comportait 331 047 faits.

Nous avons refait des expériences semblables en *PROLOG*. Les résultats ont cependant été différents. Dans ce cas, nous sommes arrivés à traiter l'ensemble des 331 047 faits que comportait notre BGC. Le programme a été consulté et interrogé en près de 6 secondes. Dans ce dernier cas, les tests ont été effectués à l'aide d'un Pentium IV avec un processeur *Intel 2.4 GHZ* accompagné de 512Mo de mémoire vive.

Voici un tableau présentant les temps nécessaires pour la vérification de la cohérence pour la propriété de symétrie pour une même base de croyance, la première en *Prolog* et la deuxième en *ASP*.

Nombre de clauses	**Temps (s) PROLOG**	**Temps (s) ASP**
30724	1.188	1.019
59320	2.266	2.057
89950	3.484	3.103
120730	4.797	3.903
149399	6.172	5.014
174654	6.671	5.830

TAB. 4.1 – Vérification de la propriété de symétrie

Les résultats obtenus lors de cette vérification de cohérence, confirment que notre problème est traitable dans des temps raisonnables en recourant aux implantations logiques retenues pour la propriété de symétrie. Ils montrent également que l'emploi du *PROLOG* ou de l'*ASP* conduisent à des résultats se rapprochant l'un de l'autre en termes de temps de vérification de cohérence dans cette situation. En fait, la vérification de cohérence s'effectue à l'intérieur de délais très brefs. L'écriture des résultats dans des fichiers textes avec *Smodels* et l'affichage à l'écran des résultats en *PROLOG* sont des tâches de durée presque plus longue que le processus de calcul conduisant à ces résultats. La vérification de la cohérence pour la propriété de symétrie des données se montre très efficace avec les deux implantations utilisées. Ainsi, il est possible de vérifier la cohérence ontologique d'un nombre important d'expressions logiques dans un délai très raisonnable autant en *PROLOG* qu'en *ASP*.

Cependant, la syntaxe de l'*ASP* est limitative en ce sens qu'elle ne permet pas d'exprimer plusieurs règles complexes de vérification de cohérence contenant des égalités et autres symboles présentées dans [MEJ04b].

À la lumière de ces résultats, nous pouvons affirmer que notre approche est efficace pour vérifier la cohérence de bases de géocroyances provenant de la BNDT pour la propriété de symétrie. Dans des travaux futurs, il serait intéressant de vérifier la cohérence de bases de géocroyances pour d'autres propriétés d'un niveau de complexité plus élevé afin de vérifier leur traitabilité.

4.6 Corrélations entre les géocroyances

Une caractéristique importante des géocroyances est la présence de corrélations fortes qui les lient souvent. Une incohérence prise isolément peut ne pas entraîner d'autres incohérences. Toutefois, une incohérence spatiale dans une base de géocroyances peut également à elle seule engendrer de grandes quantités d'incohérences. Ce type de situation se produit très souvent lors de l'ajout ou de retraits de géocroyances dans une BGC. Ce type de situation se rencontre tout autant lors de la fusion de BGC. Pour cette raison, il apparaît important de choisir de façon stratégique les géocroyances à ajouter ou à supprimer d'une BGC.

À l'occasion, le retrait ou l'ajout d'une certaine géocroyance plutôt qu'une autre qui engendre des incohérences pourra permettre d'éviter le retrait de plusieurs géocroyances qui sont devenues incohérentes entre elles. Cependant, ceci soulève la question à savoir comment déceler le ou les incohérences qui rendront possible une minimisation du nombre d'expressions logiques représentant les géocroyances incohérentes à retirer ? Une telle interrogation nous conduit à examiner des stratégies de révision de croyances géographiques. Notons que la difficulté à répondre à une telle question devient d'autant plus grande lorsqu'une BGC contient un grand nombre d'expressions logiques.

4.7 Révision des géocroyances

Lorsque des incohérences sont présentes dans des BGC, afin de pouvoir raisonner spatialement de manière rigoureuse avec des géoconnnaissances, nous devons ajouter et/ou retirer certaines d'entre elles. Cependant, nous ne sommes pas toujours prêts à éliminer toutes les informations incohérentes car ceci engendrerait des pertes d'information importantes et limiterait considérablement les interprétations possibles à partir des BGC en cause. Certaines informations apparaissent incohérentes parce que d'autres informations sont réellement incohérentes.

Il nous faut donc recourir à des connaissances dépassant le cadre de logique auquel nous faisons appel pour prendre des décisions concernant le processus de révision auquel nous nous attardons. Il devient alors essentiel de formuler des hypothèses de révision. Ces hypothèses permettent de déterminer les stratégies de révision que nous retenons afin de rétablir la cohérence de manière à répondre à nos besoins.

Le plus célèbre paradigme spécifiant le bon comportement d'opérateurs de révision est probablement celui d'Alchourron, Gardenfors et Makinson (AGM) [AGM85]. Après l'avoir examiné, nous avons constaté que les caractéristiques de ce paradigme répondaient bien à nos besoins pour la révision de géocroyances car les hypothèses émises par ces trois auteurs correspondent à des contextes typiques propres à la géomatique. En effet, en géomatique il arrive fréquemment que des bases de données géospatiales doivent être révisées à la suite d'ajouts de nouvelles informations géographiques provenant d'images montrant l'apparition et la disparition de constructions dans une ville. Ceci a un impact direct sur le contenu des bases de données géospatiales. Dans le paragraphe qui suit, nous adaptons les hypothèses des AGM à un contexte logique, soit ainsi aux BGC. Les BGC nous permettent de raisonner qualitativement de manière efficace sur des bases de données géospatiales dont ils sont une représentation qualitative.

Nous notons que le $PROLOG$ ne se prête pas bien à la révision car il ne permet pas facilement d'identifier les liens qui existent entre les atomes d'un programme et d'effectuer un choix optimal pour le retrait des atomes représentant des croyances non nécessaires pour rétablir la cohérence, ce, contrairement à l'ASP.

Afin d'effectuer une révision pertinente, nous sommes amenés à définir ce qu'est un état épistémique en géomatique et à établir des critères ou hypothèses de révision propres aux AGM. D'abord, un agent est une représentation d'une vision consensuelle d'un organisme sur les lois qui régissent une base de géocroyances. En géomatique, un état épistémique représente les croyances qu'un agent possède sur l'environnement spatial, généralement de nature géographique, reposant sur les informations qu'il détient. L'état épistémique comprend également les stratégies que l'agent utilise pour modifier ses croyances sur l'environnement géographique lors de l'ajout de nouvelles informations portant sur ce dernier.

Les critères ou hypothèses de révision propres aux AGM que nous adoptons pour la géomatique prennent la forme de postulats qui traduisent les exigences suivantes :

1. Accepter comme plus fiable la nouvelle source de géocroyances ;
2. Établir un nouvel état épistémique cohérent à la suite de l'ajout de nouvelles géocroyances ;
3. Maintenir le plus possible les anciennes croyances et ainsi minimiser les modifications des géocroyances.

Nous adoptons donc ces hypothèses des AGM pour les bases de géocroyances que nous traitons puisque que nous considérons qu'elles décrivent bien les situations habituellement rencontrées dans des BGC.

La révision des géocroyances effectuée à l'aide de la méthode des r-ensembles est une opération qui consiste à détecter le plus petit ensemble de clauses à retirer de K pour restaurer la cohérence. Dans le cas de la BNDT, les ensembles minimaux incohérents sont les plus petits ensembles de formules logiques, pouvant servir à représenter des faits, traduisant des relations entre diverses entités appartenant à la BNDT.

Dans l'exemple qui suit, nous nous concentrons sur la révision d'expressions logiques traduisant des relations de connexion. Afin de décrire notre problème, nous débutons par bien définir les variables, le domaine et les géocroyances propres à notre problème.

Variables

Les variables de notre problème sont les entités des éléments impliqués dans des relations spatiales de connexion caractérisant notre problème. Ainsi, nous définissons les variables X et Y afin de désigner les numéros d'entités impliqués dans des relations de connexion. Les cardinalités de connexions sont représentées pas les variables U et V. Ainsi, une relation de connexion entre deux entités aura la forme $c(U, V, X, Y)$. Nous désignerons par S_C l'ensemble des clauses représentant les relations entre les variables. Cet ensemble de clauses contient les règles de cohérence qui permettent de vérifier la cohérence au sein d'une base de géocroyances. Une relation de symétrie peut être définie par ArB ssi BrA.

Domaine

Le domaine propre au problème présent décrit les valeurs possibles de cardinalité de relations et les combinaisons de valeurs d'attributs possibles pour différents types de relations spatiales. Nous désignerons ce domaine par D. Les principaux types de relations rencontrés dans la BNDT sont les relations de connexion, les relations de partage et les relations d'adjacence et de superposition.

La cardinalité des relations est définie par le nombre d'instances impliquées dans la relation donnée (0,0 : aucune ; 0,1 : possibilité d'au plus une instance ; 0,2 : possibilité d'au plus deux instances ; 1,1 : Obligatoirement une et une seule instance ; 1,2 : obligatoirement une instance et possiblement deux : 2,2 : obligatoirement deux et seulement deux instances.

Le domaine est constitué de l'étendue des codes de combinaisons de valeurs d'attributs des entités de la BNDT, soient les valeurs 0 à 6000 pour les variables X et Y qui représentent ces codes. Par exemple, 361 correspond à une vanne de barrage, soit une combinaison des attributs "vanne" et "barrage". Les variables de cardinalité de relations de connexion U et V peuvent prendre les valeurs $\{0, 1, 2\}$. Nous désignerons par S_D, l'ensemble de toutes les possibilités combinatoires exprimées sous forme de clauses.

Voici quelques relations de connexion appartenant à notre problème et en l'occurrence, appartenant à notre problème, représentées sous forme de logique propositionnelle : $D = \{c(1, 1, 345, 103),\ c(1, 1, 343, 102), etc...\}$.

Géocroyances

Les géocroyances constituent l'ensemble des croyances représentées à l'aide d'expressions logiques portant sur l'environnement physique et qui sont exprimées à la lumière des informations contenues dans la BNDT. Dans le contexte de notre problème, elles correspondent aux normes et spécifications ontologiques définies dans la BNDT. Elles sont représentées sous forme de faits et de règles. Nous nommerons S_O, l'ensemble des clauses représentant les croyances géographiques traduisant les normes et spécifications de la BNDT.

Pour les fins de notre problème, nous sommes partis d'un échantillon des normes et spécifications de la BNDT initiale représentée par Mostafavi (2004) en PROLOG que nous avons traduit en *ASP*. Pour les fins de cette traduction, nous avons utilisé le traducteur que nous avons nommé *PrologASPtr*.

Représentation d'un problème de révision dans la BNDT

Dans ce qui suit, nous présentons à l'aide d'un exemple simple, un type de situation pouvant se produire dans la BNDT lors de laquelle une révision des croyances s'impose. Nous restreignons le nombre d'expressions logiques dans nos bases de géocroyances afin de faciliter l'explication de la méthode de révision des r-ensembles que nous employons. L'approche des r-ensembles appartient aux approches dites syntaxiques de la révision. Ceci signifie que la forme syntaxique d'une BGC K affecte le résultat de la révision par une BGC A. Voici un exemple simple au cours duquel nous décrivons les différentes étapes à franchir afin de réaliser une révision d'un sous-ensemble de géocroyances de la BNDT. Ainsi, nous représentons un problème de BGC incohérente en logique propositionnelle, de manière à le rendre traitable à l'aide d'implantations logiques telles que *Smodels* et *PROLOG*.

Nous débutons par la formalisation de notre problème simple qui est le suivant :

Soient K et A des ensembles de clauses finies tels que : $K \subset S_O$ où S_O désigne l'ensemble des faits contenus dans la BNDT.

Voici quelques exemples pour faciliter la compréhension des notions utilisées : $c(1, 1, 234, 235)$ signifie qu'il existe une relation de connexion de type 1-1 entre deux instances d'objets correspondants aux codes 234 et 235 de la BNDT. $c(1, 1, 234, 235)$ désigne une relation de connexion permise entre les entités 234 et 235 de la BNDT.

$c(0, 0, 234, 235)$ signifie que les relations entre les instances d'objets de codes 234 et 235 ne sont pas permises puisqu'il s'agit d'une relation de cardinalité 0-0 qui signifie une interdiction de relation. $c(0, 0, 234, 235)$ représente une interdiction de relation de connexion entre les éléments 234 et 235 de la BNDT.

Or, nous ne souhaitons pas que des situations dans lesquelles un couple permis (1-1) et un couple interdit (0-0) pour des entités identiques coexistent. Ainsi, à titre d'exemple, une situation où nous avons $c(1, 1, 235, 234) \wedge c(0, 0, 235, 234)$ constitue une incohérence spatiale dans la BNDT.

L'expression logique $c(1, 1, 235, 234) \wedge c(0, 0, 235, 234)$ est équivalente à la clause $\neg c(1, 1, 235, 234) \vee \neg c(0, 0, 235, 234)$

$\neg c(1, 1, 235, 234) \vee \neg c(0, 0, 235, 234)$ se traduit par les règles suivantes :

$\neg c(1, 1, 235, 234) \rightarrow c(0, 0, 235, 234)$ et
$c(1, 1, 235, 234) \leftarrow \neg c(0, 0, 235, 234)$

Ainsi, pour notre problème nous définissons une base de croyances K telle que :

$K = \{c(1, 1, 234, 235), c(1, 1, 235, 234), c(1, 1, 237, 238), c(1, 1, 238, 237)\}$

que nous cherchons à réviser par de la nouvelle information contenue dans A :

$A = \{c(0, 0, 234, 235), c(0, 0, 235, 234), \neg c(0, 0, 234, 235) \vee \neg c(1, 1, 234, 235),$
$\neg c(1, 1, 235, 234) \vee \neg c(0, 0, 235, 234)\}$.

Soit V l'ensemble des atomes naturels et artificiels de $K \cup A$ tel que :
$V = \{c(1, 1, 234, 235), c(1, 1, 235, 234), c(1, 1, 237, 238), c(1, 1, 238, 237), c(0, 0, 234, 235),$
$c(0, 0, 235, 234), c(1, 1, 234, 235)', c(1, 1, 235, 234)', c(1, 1, 237, 238)', c(1, 1, 238, 237)',$
$c(0, 0, 234, 235)', c(0, 0, 235, 234)', r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r_{c(1,1,238,237)},$
$r'_{c(1,1,234,235)}, r'_{c(1,1,235,234)}, r'_{c(1,1,237,238)}, r'_{c(1,1,238,237)}\}$

Nous définissons R_K, l'ensemble des atomes de V de la forme r_c ou r'_c. Ainsi,
$R_K = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r_{c(1,1,238,237)},$
$r'_{c(1,1,234,235)}, r'_{c(1,1,235,234)}, r'_{c(1,1,237,238)}, r'_{c(1,1,238,237)}\}$.

Nous définissons
$R_K^+ = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r_{c(1,1,238,237)}\}$
l'ensemble des atomes originaux de R_K, et
$R_K^- = \{r'_{c(1,1,234,235)}, r'_{c(1,1,235,234)}, r'_{c(1,1,237,238)}, r'_{c(1,1,238,237)}\}$
l'ensemble des atomes artificiels de R_K .

Nous visons à réviser K par A. Voici les trois étapes de construction du programme logique $P_{K \cup A}$ afin d'appliquer la méthode de calcul des r-ensembles pour la révision :

1) Nous introduisons des règles de manière à ce que les modèles stables de $P_{K \cup A}$ entrent en bijection avec les interprétations de V^+. Ainsi, $\forall a$ de V^+, afin de prendre en compte la négation classique de certains atomes, nous introduisons les règles $a \leftarrow not\ a'$ et $a' \leftarrow not\ a$.

2) Nous introduisons des règles pour empêcher que les interprétations I qui ne sont pas des modèles de A de correspondre à des modèles stables. Ainsi, pour toute clause c de A telle que c= $\neg a_0 \vee ... \vee \neg a_n, a_{n+1}, ..., a_m$, nous ajoutons la règle suivante :

$false \leftarrow a_0, ..., a_n, a'_{n+1}, ..., a'_m$

Au programme, pour éliminer l'atome *false* des modèles stables recherchés, nous ajoutons la règle :

$contradiction \leftarrow false, not\ contradiction$

ce qui permet de faire en sorte que les modèles stables de $P_{K\cup A}$ correspondent aux interprétations de V^+ qui sont des modèles de A.

3) Nous sélectionnons les interprétations de I qui correspondent à des modèles stables de $P_{K\cup A}$ et qui sont des modèles de $(K \cup A) \setminus Cls(I \cap R^+_K)$. Pour ce faire, pour chaque clause c de K tel que $c = \neg b_0 \vee ... \vee \neg b_n, b_{n+1}, ..., b_m$, nous introduisons la règle suivante : $r_c \leftarrow b_0, ..., b_n, b'_{n+1}, ..., b'_m$.

Nous exprimons notre problème en suivant les trois étapes décrites ci-dessus. Le problème logique obtenu à la lumière de ces trois étapes est le suivant :

$c(1, 1, 234, 235) \leftarrow not\ c(1, 1, 234, 235)'$
$c(1, 1, 235, 234) \leftarrow not\ c(1, 1, 235, 234)'$
$c(1, 1, 237, 238) \leftarrow not\ c(1, 1, 237, 238)'$
$c(1, 1, 238, 237) \leftarrow not\ c(1, 1, 238, 237)'$
$c(0, 0, 234, 235) \leftarrow not\ c(0, 0, 234, 235)'$
$c(0, 0, 235, 234) \leftarrow not\ c(0, 0, 235, 234)'$

$c(1, 1, 234, 235)' \leftarrow not\ c(1, 1, 234, 235)$
$c(1, 1, 235, 234)' \leftarrow not\ c(1, 1, 235, 234)$
$c(1, 1, 237, 238)' \leftarrow not\ c(1, 1, 237, 238)$
$c(1, 1, 238, 237)' \leftarrow not\ c(1, 1, 238, 237)$
$c(0, 0, 234, 235)' \leftarrow not\ c(0, 0, 234, 235)$
$c(0, 0, 235, 234)' \leftarrow not\ c(0, 0, 235, 234)$

$r_{c(1,1,234,235)} \leftarrow not\ r'_{c(1,1,234,235)}$
$r_{c(1,1,235,234)} \leftarrow not\ r'_{c(1,1,235,234)}$
$r_{c(1,1,237,238)} \leftarrow not\ r'_{c(1,1,237,238)}$
$r_{c(1,1,238,237)} \leftarrow not\ r'_{c(1,1,238,237)}$
$r'_{c(1,1,234,235)} \leftarrow not\ r_{c(1,1,234,235)}$
$r'_{c(1,1,235,234)} \leftarrow not\ r_{c(1,1,235,234)}$
$r'_{c(1,1,237,238)} \leftarrow not\ r_{c(1,1,237,238)}$
$r'_{c(1,1,238,237)} \leftarrow not\ r_{c(1,1,238,237)}$

$false \leftarrow c(0,0,234,235)'$
$false \leftarrow c(0,0,235,234)'$
$false \leftarrow c(0,0,234,235), c(1,1,234,235)$
$false \leftarrow c(0,0,235,234), c(1,1,235,234)$
$contradiction \leftarrow false,\ not\ contradiction$

$rc(1,1,234,235) \leftarrow c(1,1,234,235)'$
$rc(1,1,235,234) \leftarrow c(1,1,235,234)'$
$rc(1,1,237,238) \leftarrow c(1,1,237,238)'$
$rc(1,1,238,237) \leftarrow c(1,1,238,237)'$

En procédant par négation par échec, *Smodels* navigue à l'intérieur d'un arbre de preuves afin d'identifier des solutions. Le problème ci-dessus est instancié avec *Lparse* et les modèles obtenus avec *Smodels* sont les suivants :

$S_1 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r'_{c(1,1,237,238)}, r_{c(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234)', c(1,1,238,237)', c(1,1,237,238), c(0,0,234,235), c(0,0,235,234)\}$

$S_2 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r_{c(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234)', c(1,1,238,237)', c(1,1,237,238), c(0,0,234,235), c(0,0,235,234)\}$

$S_3 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r_{c(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234)', c(1,1,237,238)', c(1,1,238,237)', c(0,0,234,235), c(0,0,235,234)\}$

$S_4 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r'_{cp(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234)', c(1,1,237,238)', c(1,1,238,237), c(0,0,234,235), c(0,0,235,234)\}$

$S_5 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r_{c(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234)', c(1,1,237,238)', c(1,1,238,237), c(0,0,234,235), c(0,0,235,234)\}$

$S_6 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r_{c(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234)', c(1,1,237,238), c(1,1,238,237), c(0,0,234,235), c(0,0,235,234)\}$

$S_7 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r'_{c(1,1,237,238)}, r_{c(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234)', c(1,1,237,238), c(1,1,238,237), c(0,0,234,235), c(0,0,235,234)\}$

$S_8 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r'_{cp(1,1,237,238)}, r'_{cp(1,1,238,237)}, c(1,1,234,235)',$
$c(1,1,235,234), c(1,1,237,238)', c(1,1,238,237), c(0,0,234,235), c(0,0,235,234)\}$

$S_9 = \{r_{c(1,1,234,235)}, r_{c(1,1,235,234)}, r_{c(1,1,237,238)}, r'_{c(1,1,238,237)}, c(1,1,234,235)',$

$c(1,1,235,234)', c(1,1,237,238), c(1,1,238,237), c(0,0,234,235), c(0,0,235,234)\}$

À ce stade, à partir des modèles que nous venons de calculer, nous devons identifier quels sont les ensembles R que nous pourrions retirer afin de rétablir la cohérence au sein de $K \cup A$. Calculons maintenant les résultats de Cls appliquée aux intersections entre les neuf modèles obtenus et R_K^+ :

$$Cls(S_1 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,238,237)\}$$
$$Cls(S_2 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238), c(1,1,238,237)\}$$
$$Cls(S_3 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238), c(1,1,238,237)\}$$
$$Cls(S_4 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238)\}$$
$$Cls(S_5 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238), c(1,1,238,237)\}$$
$$Cls(S_6 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238), c(1,1,238,237)\}$$
$$Cls(S_7 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,238,237)\}$$
$$Cls(S_8 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234)\}$$
$$Cls(S_9 \cap R_K^+) = \{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238)\}$$

$$Cls(S_1 \cap R_K^+) = Cls(S_7 \cap R_K^+) = R_1$$
$$Cls(S_2 \cap R_K^+) = Cls(S_3 \cap R_K^+) = Cls(S_5 \cap R_K^+) = Cls(S_6 \cap R_K^+) = R_2$$
$$Cls(S_4 \cap R_K^+) = Cls(S_9 \cap R_K^+) = R_3$$
$$Cls(S_8 \cap R_K^+) = R_4$$

Dans le cas présent, nous avons utilisé *Smodels* afin d'obtenir les modèles stables de notre problème et nous avons calculé manuellement les différentes valeurs possibles de R, issues de l'application de Cls sur l'intersection des modèles stables R_K^+. Il serait intéressant d'utiliser l'adaptation de *Smodels* développée par Papini et Würbel [Pa03] pour la révision afin de ne pas avoir à calculer manuellement ces intersections dans une situation comme celle qui est propre à notre problème. De plus, l'adaptation de *Smodels* permet d'obtenir directement le ou les r-ensemble(s) que nous cherchons, et ce, en plus d'élaguer efficacement l'arbre de recherche de modèles stables du problème étudié. Nous pourrions calculer les R_K-*générateurs* avec l'adaptation de *Smodels* pour ensuite pouvoir obtenir les r-ensembles directement. Toutefois, l'utilisation d'une telle adaptation dépasse le cadre de ce mémoire. En effet, l'utilisation d'un tel outil nécessiterait le développement d'un générateur de règles d'ASP automatisé afin d'être en mesure de réviser les bases de croyances que nous avons traitées pour la vérification de cohérences.

En utilisant *Smodels* et en calculant les différents ensembles R, nous obtenons que les différents ensembles R découlant des résultats de notre démarche sont :

$R_1 = \{c(1,1,234,235), c(1,1,235,234), c(1,1,238,237)\}$,
$R_2 = \{\{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238), c(1,1,238,237), \}$,
$R_3 = \{c(1,1,234,235), c(1,1,235,234), c(1,1,237,238)\}$ et
$R_4 = \{c(1,1,234,235), c(1,1,235,234)\}$.

Le r-ensemble, soit l'ensemble d'atomes à retirer est $R_4 = \{c(1,1,234,235), c(1,1,235,234)\}$ étant donné qu'il s'agit du seul ensemble R qui respecte l'hypothèse du changement minimal et qui en même temps ne brise pas la symétrie entre les atomes. Ainsi, en retirant R_4, seulement deux atomes sont retirés soient $c(1,1,234,235)$ et $c(1,1,235,234)$ et la symétrie entre les atomes restant de la base K soient $c(1,1,237,238)$ et $c(1,1,238,237)$ est maintenue. Retirer R_1, R_2 ou R_3 aurait signifié extraire respectivement 3, 4 ou 3 atomes ce qui n'aurait pas permis de respecter l'hypothèse du changement minimal contrairement au retrait de R_4 qui permet de la respecter. De plus le retrait de R_1 ou R_3 aurait brisé la propriété de symétrie et le retrait de R_2, nous aurait conduit à obtenir une BGC K vide, ce qui aurait été dépourvu d'intérêt pour la solution de notre problème. Ainsi, R_4 est le r-ensemble que nous cherchons. Ainsi, en retirant l'ensemble de clauses R_4 de $K \cup A$, nous rétablissons efficacement la cohérence au sein de la BGC K révisée par A.

Comparaison entre les langages de programmation *ASP* et *PROLOG*

Les langages *ASP* et *PROLOG* se montrent adéquats pour les recherches d'incohérences dans une base de géocroyances formée à partir de la BNDT. Nous observons également qu'une série de similarités et de différences existent entre les deux langages, ce que le tableau suivant résume :

	PROLOG	**ASP (Smodels)**
Types de clauses traités	Horn	Tous les types
Concepts fondamentaux	Unification, récursivité, retour sur trace, résolution du 1er ordre	Unification, récursivité, retour sur trace, SAT, modèles stables
Ordre des expressions logiques	Avec conséquences	Sans conséquences
Type de raisonnement	monotone ou non monotone	monotone ou non monotone
Négation	Hypothèse du monde clos	Hypothèse du monde clos
Indétermination d'un prédicat dans la partie NOT d'une règle	Non tolérée	La règle s'applique et les prédicats de la partie NOT sont considérés faux
Traitement des listes	présent	absent
Exécution	Réalisation de requêtes	Recherche de modèles stables

Tab. 4.2 – PROLOG vs ASP

Lors de la recherche de solution en *PROLOG* dans des programmes élémentaires de *PROLOG*, l'ordre des expressions logiques n'a pas d'influence sur le résultat de requêtes si l'arbre de recherche ne contient pas de branches infinies. La situation diffère cependant lorsque des bases de croyances sont enrichies, entre autres, par de nouveaux faits ou réduites par le retrait de faits avec les commandes *asserts* et *retract*. Toutefois, l'ordre des expressions logiques peut avoir un impact sur le temps nécessaire pour trouver des solutions dans tout programme, ce, autant en *ASP*, qu'en *PROLOG*.

Les limites du *PROLOG* sont plus apparentes dans des contextes de révision. Programmer une approche de révision telle que celle des r-ensemble en *PROLOG* s'annonce très compliqué et lourd, entre autres, parce que le concept de modèle stable n'existe pas dans la forme originale du langage *PROLOG*. Ainsi, en *PROLOG*, identifier stratégiquement les ensembles minimaux de clauses à retirer d'une BGC se présente comme un problème ardu. Une telle approche n'aurait pas la même élégance que la méthode des r-ensembles qui a été implantée en *ASP*. En *PROLOG*, si les commandes *asserts* et *rectract* sont utilisées, rien ne garantit que la BGC modifiée par le retrait de croyances préservera sa cohérence. Ceci est particulièrement problématique avec les BGC traitées en *PROLOG*. Les possibilités supplémentaires que présentent l'ASP sont un atout considérable. Cette réalité explique que des chercheurs tels que Omar Elkhatib, Enrico Pontelli et Tran Cao Son aient développé un système logique permettant de bénificier des fonctionnalités intéressantes de l'*ASP* en *PROLOG* dans *ASP-PROLOG : A System for Reasoning about Answer Set Programs in Prolog*(2004).

4.8 Bilan de chapitre

Nous avons constaté que les bases de géocroyances se prêtent bien au raisonnement spatial dans des contextes tels que ceux rencontrés en géomatique. À la lumière des résultats obtenus, nous avons vu que le *PROLOG* est un langage de programmation logique utile pour la vérification de cohérence de BGC mais limité sur le plan des opérations de révisions de BGC. Nous avons vu finalement que la programmation ASP s'applique bien à la résolution du problème de vérification de cohérence tout comme à celui de la révision de BGC.

Chapitre 5

Conclusion

Dans le cadre de ce mémoire, nous avons effectué une comparaison d'une approche de raisonnement qualitatif basée sur le *PROLOG* avec une autre reposant sur la l'*ASP* dans un contexte spatial. La principale question de recherche à laquelle nous tentions de répondre était si une implantation du raisonnement non monotone faisant intervenir le concept de modèle stable pouvait nous permettre de résoudre des problèmes de vérification de cohérence et des problèmes de révision dans un contexte géographique donné. Nous avons mis en lumière les résultats de ces comparaisons d'abord dans un contexte de vérification de cohérence d'une base de géocroyances (BGC) et ensuite dans celui de la révision des géocroyances d'un échantillon de la BNDT.

Dans le cadre de ce travail, nous visions à atteindre plusieurs objectifs. Premièrement, nous souhaitions définir et créer des BGC en formalisant des ontologies et en effectuant un choix de mode de représentation des croyances adapté au contexte spatial. Nous voulions vérifier la cohérence dans une BGC à la suite de l'ajout de nouvelles informations en suivant une approche ontologique. Nous visions également à expérimenter des retraits d'incohérences à l'intérieur d'une BGC dans des contextes spatiaux simples. Nous souhaitions de plus effectuer une comparaison d'une approche de raisonnement spatial non monotone implantée en *PROLOG* et une seconde en *ASP* faisant intervenir le concept de modèles stables.

Notre démarche, adoptée dans le but de l'atteinte de ces objectifs, nous a conduit à suivre deux grandes phases de recherche. La première a consisté à explorer les limites de différentes logiques dans un contexte de programmation pour différentes situations de raisonnement spatial. Au cours de cette phase, nous avons exploré les limites de la logique classique ainsi que celles des logiques non monotones et avons constaté que des implantations respectives de ces deux logiques nous permettaient de raisonner en respectant les caractéristiques de ces deux grands types de logique. À la lumière des travaux de Mostafavi et Edwards (2004), nous avons étudié le fonctionnement du raisonnement monotone à l'aide d'un moteur de raisonnement de *PROLOG*. Nous avons ensuite examiné comment une approche de raisonnement non monotone pouvait s'appliquer au problème de vérification de cohérence et à celui de la révision des bases de géocroyances. Cette phase nous a permis d'établir les bases théoriques (Chapitre 2) nécessaires pour ce travail.

Par la suite, la deuxième phase consistait à appliquer et à comparer au contexte de la BNDT les deux approches de raisonnement retenues, soit dans un premier temps une méthode utilisant le *PROLOG* et dans un deuxième temps, une méthode de raisonnement utilisant l'*ASP*.

Pour ce faire, nous avons défini un contexte spatial d'application de la logique non monotone propre à la BNDT, soit celui de différentes relations de connexion provenant de la BNDT. Nous avons défini des BGC et détaillé leurs principales caractéristiques, ce qui a été l'objet du Chapitre 3. Nous sommes partis de la représentation des connaissances géographiques de la BNDT proposée par Mostafavi et Edwards (2004) en *PROLOG* que nous avons simplifiée afin de tenir compte des contraintes représentationnelles propres à l'*ASP* et de proposer des représentations équivalentes en *PROLOG* et en *ASP* pour les fins de nos comparaisons. Nous avons créé un traducteur de code *PROLOG* en *ASP*. L'utilisation de ce traducteur nous a permis d'obtenir des BGC contenant des expression logiques équivalentes. Or, nous avons réorganisé ces BGC de manière à respecter le concept de BGC développé au Chapitre 3. Nous avons validé la cohérence dans une BGC à la suite de l'ajout de nouvelles informations en recourant à une approche ontologique suivant une approche classique en *PROLOG* et ensuite en nous appuyant sur une approche faisant intervenir le concept de modèles stables. Finalement, nous avons exploré l'application de la méthode des r-ensembles à l'intérieur d'une BGC dans un contexte de révision.

Notre démarche nous a conduit à l'obtention d'un certain nombre de résultats. Il est apparu qu'il importait d'avoir une emprise sur les géocroyances que nous cherchions à modéliser afin d'arriver à constituer un bon modèle de croyances qui nous permet de réaliser une formalisation rigoureuse des géocroyances. Ceci a été nécessaire afin de pouvoir créer une représentation logique adéquate qui serait traitable par les moteurs de raisonnement logique. Ainsi, toutes les expressions logiques n'étaient pas directement traitables par les moteurs de raisonnement logique. Cette recherche nous a mené à réaliser que les implantations d'approches logiques ne faisant pas intervenir le concept de modèles stables sont toutes aussi pertinentes que celles le faisant intervenir si ce n'est à quelques points près lorsque nous travaillons sur le problème de vérification de cohérence. Toutefois, l'outil *Smodels* est loin d'offrir les mêmes possibilités syntaxiques que le *PROLOG* : L'*ASP* ne permet pas d'utiliser le concept de listes et certaines formes d'égalités permises en *PROLOG*. Ainsi, il nous a été impossible d'effectuer certains tests de vérifications de cohérence, réalisés par Mostafavi et Edwards [MEJ04b] en *PROLOG*, en *ASP*. Dans les deux cas, soit en *ASP* et en *PROLOG*, les implantations utilisées reposaient sur de la négation avec échec. Une différence se situait au niveau du type de réponse recherché. Dans le premier cas, il s'agissait de modèles stables et dans le deuxième cas de réponses à des requêtes. Nous comprenons que rechercher une série d'ensembles solutions (modèles) est généralement plus complexe que de simplement répondre à une requête unique. Malgré cette réalité, dans notre cas pour la vérification de la cohérence symétrique des relations de connexion, il n'y avait qu'un seul modèle stable (ASP) et une requête sur la symétrie en *PROLOG*. Ceci a fait en sorte que les calculs n'étaient pas d'une trop grande complexité. Ainsi, nous avons obtenu des temps presque semblables pour cette opération, qu'il s'agisse du moteur de raisonnement *Smodels* ou de celui de *PROLOG*. Ces temps ont été compilés à l'intérieur d'un tableau au Chapitre 4.

Pour la révision de géocroyances les constats diffèrent. En fait, les cas de révisions de géocroyances moindrement complexes deviennent pratiquement intraitables en *PROLOG*, car la notion de modèle stable n'existe pas directement dans ce langage, ce que nous avons constaté. Il existe cependant l'*ASP-PROLOG* une extension du *PROLOG* auquel les possibilités de l'*ASP* sont ajoutées. Ainsi, il est pratiquement impossible de cerner les multiples ensembles d'expressions logiques dépendantes correspondant à des incohérences en *PROLOG*. Pour cette raison, l'utilisation de l'*ASP* combinée à la méthode des r-ensembles est nettement plus intéressante et permet de traiter des cas complexes de révision et de cerner les dépendances entre les expressions logiques. L'*ASP-PROLOG* apparaît comme une avenue de recherche intéressante permettant de bénéficer de la syntaxe étendue du *PROLOG* en *ASP*.

Nous avons constaté que l'implantation de la révision de géocroyances par la méthode des r-ensembles ne peut être entièrement complétée avec l'usage de la version originale de *Smodels*. Avec *Smodels*, si le calcul des modèles stables peut être effectué moyennant des coûts importants en termes de temps, puisque l'ensemble des modèles stables y est calculé, néanmoins les r-ensembles ne peuvent y être calculés directement. Avec cette méthode, nous avons résolu manuellement un problème de révision simple exposé dans le chapitre 4. Cette procédure ouvre une porte à des travaux futurs sur l'utilisation de l'adapatation de Smodels [Pa03] pour le calcul des r-ensembles correspondant directement aux incohérences à retirer de la BNDT sur des BGC plus volumineuses et d'un niveau de complexité plus élevé. Il pourrait alors être intéressant d'étudier les possibilités et les limites de traitabilité liées à l'utilisation de cette adaptation dans le contexte de la BNDT. Il y aurait lieu également de voir, si pour ce type de problème des heuristiques particulières intégrées à l'adapatation de *Smodels*, reposant sur les caractéristiques de notre problème portant sur la BNDT, pourraient apporter des gains en termes d'efficacité.

Il reste une question importante à laquelle il serait intéressant de répondre : Est-il possible de cerner les dépendances entre les incohérences (modèles stables) autrement qu'avec des méthodes logiques numériques, par exemple ?

Bibliographie

[AGM85] C. E. Alchourron, P. Gärdenfors, and D. Makinson. On the logic of theory change : Partial meet contraction and revision functions. In *Journal of symbolic Logic*, pages 510–530, 1985.

[BBJ+05] Salem BENFERHAT, J. Bennaim, R. Jeansoulin, M. Khelfallah, Sylvain LAGRUE, O. Papini, N. Wilson, and E. Wurbel. Belief revision of gis systems : the results of revigis. In Lluis Godo (ed.), editor, *8th European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty(ECSQARU'05)*, pages 452–464. LNAI3571, Springer, juillet 2005.

[BE01] T. E. Bittner and G. Edwards. Towards an ontology for geomatics. *Journal of the Canadian Institute of Geomatics, Geomatica*, 55(4) :475–490, 2001.

[BLP05] S. Benferhat, S. Lagrue, and O. Papini. Revision of partially ordered information. In *proceedings of International Joint Conference on Artificial Intelligence IJCAI '05*, pages 376–381, August 2005.

[BNBPW04] J. Ben-Naim, S. Benferhat, O. Papini, and E. Würbel. *An Answer Set Programming Encoding of Prioritized Removed Sets Revision : Application to GIS*, volume 3229/2004. Springer Berlin / Heidelberg, 2004.

[Bro04] J. Brodeur. *Interopérabilité des données géospatiales : Élaboration du concept de proximité géosémantique.* Thèse de doctorat, Université Laval, Québec, 2004.

[CIT] CITS. Bndt. http ://www.cits.rncan.gc.ca/.

[Coh96] A. G. Cohn. Calculi for qualitative spatial reasoning. In Jacques Calmet, J. Campbell, and J. Pfalzgraf, editors, *Artificial Intelligence and Symbolic Mathematical Computation*, pages 124–143, Berlin, 1996. Springer-Verlag.

[Dal88] M. Dalal. Investigations into a theory of knowledge base revision : Preliminary report. In *In Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'88)*, pages 475–479, 1988.

[dgLS89] Membres du groupe Léa Sombé. *Raisonnements sur des informations incomplètes en intelligence artificielle*. Teknea, Toulouse , France, 1989.

[DP97] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artifical Intelligence*, 89(1-2) :1–29, 1997.

[EH90] M. J. Egenhofer and J. R. Herring. Categorizing binary topological relationships between regions. Technical report, Department of Surveying Engineering, University of Maine Orono Maine, 1990.

[Gac97] L. Gacogne. *Eléments de logique floue*. Hermès, Paris, 1997.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[GLP07] F. Leber G. Ligozat and O. Papini. *L'espace et le temps en intelligence artificielle : modèles de raisonnement et applications (à paraître)*. Hermes, 2007.

[Gru93a] T. R. Gruber. *Formal ontology in conceptual analysis and knowledge representation, Chapter : Towards principles for the design of ontologies used for knowledge sharing*. Kluwer Academic Publishers, 1993.

[Gru93b] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, 1993.

[GS04] P. Grenon and B. Smith. Snap and span : Towards dynamic spatial ontology. *Spatial Cognition and Computation*, 4(1) :69–104, 2004.

[HA28] D. Hilbert and W. Ackermann. *Principles of Mathematical Logic*. Springer-Verlag, 1928. Traduction vers l'anglais de Lewis Hammond et al. (1950).

[Lag03] S. Lagrue. *Gestion d'informations partiellement ordonnées : raisonnement, révision et information géographique*. PhD thesis, Université du Sud - Toulon - Var, Toulon, France, Décembre 2003.

[Mar58] R. Martin. *Les fondements de la logique*. Encyclopédie Universalis 10 électromique. Gallimard, Paris, 1958.

[MEJ04a] M.-A. Mostafavi, G. Edwards, and R. Jeansoulin. A formal method for quality assesment of spatial data bases : The role of ontologies. In *Joint meeting of TIES and the Spatial Accuracy Symposium*, Portland, Maine, USA, June 28 - July 1 2004.

[MEJ04b] M.-A. Mostafavi, G. Edwards, and R. Jeansoulin. An ontology-based method for quality assesment of spatial data bases. In *Actes de conférence de ISSDQ 2004*, Autriche, 15 au 17 avril 2004.

[MHV05] S. Munoz-Hernandez and C. Vaucheret. Extending prolog with incomplete fuzzy information, 2005.

[Nor99] P. Normand. *Modélisation des contraintes d'intégrité spatiales : Théorie et exemples d'application*. Mémoire de maîtrise, Université Laval, Québec, 1999.

[NS97] I. Niemelä and P. Simons. Smodels - an implementation of the stable model and well-founded semantics for normal logic programs. In *Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning*, volume 1265, pages 420–429, Dagstuhl, Germany, July 1997.

[Pa03] O. Papini and al. Rev !gis : Uncertain geographic knowledge, maintenance and revision. Technical report, Université du SUD Toulon-Var, May 2003.

[Pap92] O. Papini. A complete revision function in propositional calculus. In B. Neumann, editor, *Proceedings of ECAI92*, pages 339–343. John Wiley and Sons. Ltd, 1992.

[Pap00] O. Papini. Knowledge base revision. *Knowl. Eng. Rev.*, 15(4) :339–370, 2000.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

[RCC92] D. A. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In B. Nebel, C. Rich, and W. Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning : Proceedings of the Third International Conference*, pages 165–176, San Mateo, California, 1992.

[Rei87] R. Reiter. A logic for default reasoning. In M. L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 68–93, Los Altos, Caifornie, États-Unis, 1987.

[Rob79] J.A. Robinson. *Logic : Form and Function*. Edinburgh University Press, 1979.

[Rob94] *Le petit Robert, Dictionnaire alphabétique et analogique de la langue française*. Paris, 1994.

[SAA+00] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, and B. Wielinga. *Knowledge Engineering and Management*. The MIT Press, Cambridge (Massachussets), 2000.

[Sch97] K. Schlechta. *Nonmonotonic logics Basic Concepts, Results, and Techniques*. Springer Lecture Notes series. Springer-Verlag, Berlin, Heidelberg, New york, 1997.

[SM01] B. Smith and D. Mark. Geographical categories : An ontological investigation. *International Journal of Geographical Information Science*, 15(7) :591–612, 2001.

[Spo88] W. Spohn. Ordinal conditional functions : A dynamic theory of epistemic states. In *Causation in Decision, belief change and Statistics*, pages 105–134, 1988.

[WPJ00] E. Wurbel, O. Papini, and R. Jeansoulin. Revision : An application in the framework of gis. In *7th International Conference on Principles of Knowledge Representation and Reasoning, KR'2000*, pages 505–516, Breckenridge, Colorado, USA, avril 2000.

[WR12] A.N. Whitehead and B. Russell. *Principia Mathematica*, volume 2. Cambridge University Press, 1912.

[Wur00] E. Wurbel. *Révision de connaissances géographiques*. Doctorat informatique, Université de Provence, Marseille, 22 décembre 2000.