

Logiques modales et bases de données

Laurence Cholvy — Frédéric Cuppens — Robert Demolombe

ONERA-CERT
2 avenue Ed. Belin
31055 Toulouse Cedex

RÉSUMÉ. Cet article a pour objet de présenter les logiques modales comme outil de modélisation des bases de données. Après une rapide introduction aux logiques modales, nous présentons trois exemples de modélisation. Le premier problème est de déterminer pour une question donnée, les parties de la réponse qui sont valides ou complètes vis-à-vis du monde réel. Le deuxième problème est relatif à la fusion de données contradictoires. Le troisième concerne les bases de données sécurisées.

ABSTRACT. This paper aims to present modal logics as a tool for database modelling. We first give a brief introduction to modal logics. Then, we present three examples of modelling. In the first one, the problem is, given a query addressed to a database, to characterize the parts of the answer which are valid or complete in regard to the real world. The second problem is to query a database obtained from merging several contradictory databases. Finally, the third problem concerns modelling secure databases.

MOTS-CLÉS : Logique Modale, Bases de Données, Systèmes d'Information, Modélisation.

KEY WORDS : Modal Logics, Data Bases, Information Systems, Modelling.

1. Introduction

Au cours des années 70, et jusqu'au début des années 80, est apparu un important courant de recherche qui avait pour objectif de formaliser en logique un certain nombre de problèmes rencontrés dans le domaine des bases de données. Cette approche, appelée "Logique et Bases de Données"[GAL 78, GAL 81, GAL 83, GAL 84], a mis quelque temps avant d'être acceptée, à une époque où les concepts utilisés dans les bases de données n'étaient pas toujours clairement dissociés des considérations technologiques. Actuellement l'intérêt de cette démarche est bien reconnue dans la communauté scientifique.

Depuis les années 80, la notion de base de données s'est élargie aux bases de données déductives, aux bases de connaissances, et aux bases de données distribuées. De plus, les problèmes de bases de données sont souvent analysés dans le contexte plus large des systèmes d'informations. Avec ces extensions sont apparues de nouvelles

problématiques pour lesquelles les logiques modales sont mieux appropriées que les logiques classiques, et parfois même sont indispensables.

Par exemple, pour modéliser les contraintes d'intégrité portant sur plusieurs états d'une base, et pour raisonner sur ces contraintes, les logiques temporelles ont été utilisées dès le début des années 80 [CAS 82b], [CAS 82a], [CAS 84], [KUN 84], [KUN 85], [CHO 86b], [CHO 86a]. Plus récemment, le problème de la génération automatique, à partir de contraintes temporelles, de séquences de mises à jour qui ne violent pas les contraintes, a été abordé [BID 93], [BID 95], [AMO 95]. D'une façon plus générale la notion de contrainte d'intégrité en elle-même a pu être mieux définie en distinguant, grâce à la logique épistémique [REI 88], [REI 92], [DEM 96d] et à la logique déontique [DEM 96c], des contraintes qui portent sur le contenu de la base elle-même, et d'autres qui portent sur les objets du domaine d'application.

Un autre exemple concerne les bases de données distribuées pour lesquelles il est nécessaire de préciser dans quel contexte une information est vraie. En effet, il peut se faire que des bases différentes contiennent des informations contradictoires. En logique classique on n'a pas la possibilité de distinguer différents contextes, par contre les logiques doxastiques et épistémiques permettent de le faire très naturellement.

Citons encore l'exemple de la formalisation des réglementations portant sur la confidentialité des informations, qui définissent ce que les usagers ont la permission ou l'interdiction de connaître. Les travaux en logiques déontiques proposent une grande variété de techniques, plus ou moins sophistiquées, pour formaliser les réglementations [BON 92], [BON 95].

Enfin pour concevoir des systèmes coopératifs [CUP 88, CAZ 92, DEM 94a] qui fournissent des réponses aux questions contenant, en plus des informations explicitement demandées, des informations complémentaires pertinentes, on doit formaliser la notion de pertinence. Une modélisation dans une logique modale particulière a été définie qui en donne une première approximation [DEM 95, DEM 96e]. Celle-ci permet de raisonner sur les thèmes auxquels se rapportent les informations, et sur les thèmes auxquels s'intéressent les usagers.

Sur ces différents problèmes des résultats très significatifs ont déjà été acquis, mais il faut bien préciser où se situe l'intérêt de la modélisation en logique modale. Le principal intérêt, comme pour les logiques classiques, est de proposer des formalismes qui peuvent être utilisés pour modéliser les notions rencontrées dans divers types de problèmes. Au moment d'exprimer les problèmes en logique on est forcé de s'interroger sur le sens qu'on veut donner à telle ou telle notion. Le résultat de la modélisation est une meilleure compréhension des problèmes, et l'efficacité, dans tous les sens du terme, de l'implémentation d'une solution à un problème, sera d'autant meilleure que le problème est mieux compris, et mieux formalisé.

Le deuxième intérêt est que le résultat de la modélisation est exprimé dans un langage formel dont le traitement peut être automatisé sous forme de maquette¹. Une maquette joue un rôle important comme outil de modélisation, car elle permet de véri-

1. Il existe de tels outils de maquettage pour certaines logiques modales tels que MOLOG [FAR 85]. D'autres techniques, très générales, consistent à définir des traductions automatiques des logiques modales vers les logiques classiques, puis à utiliser des outils tels que PROLOG.

fier, sur de petits exemples caractéristiques, si les conséquences que l'on peut déduire de ces exemples correspondent à ce qui est attendu. Si ce n'est pas le cas, on devra modifier la logique utilisée pour la modélisation jusqu'à obtenir une modélisation qui soit jugée acceptable. Il faut noter également que raisonner "à la main" sur des exemples, même très simples, qui requièrent l'imbrication de plusieurs modalités, s'avère rapidement inextricable.

Enfin, le troisième intérêt de la modélisation en logique est de fournir un bon point de départ pour définir des stratégies de traitement efficaces. En effet ce type de modèles n'est pas inutilement compliqué par des notions relatives à telle ou telle technologie qui relèvent de choix faits a priori. De plus, ils sont tout à fait neutres vis-à-vis des stratégies de dérivation et laissent toute liberté de choix aux concepteurs qui cherchent à les optimiser.

Le but de cet article est de faire mieux connaître les logiques modales et de montrer leur intérêt comme outil de modélisation. Dans la section 2 nous présentons une brève introduction aux logiques modales qui est à la fois rigoureuse et qui peut se comprendre intuitivement. Ensuite, nous présentons trois exemples de modélisation relatifs, aux problèmes de la fiabilité des sources d'informations qui insèrent des données dans une base (section 3), à la fusion de données contradictoires (section 4), et à la réglementation de la confidentialité (section 5). Dans chacun des cas, nous partons d'une analyse informelle du problème, nous décrivons la logique modale appropriée, puis nous illustrons son utilisation sur un exemple.

2. Introduction aux logiques modales

L'objectif des logiques formelles est de donner un sens précis aux énoncés d'un langage utilisé pour représenter le monde², et de définir les règles qui permettent de raisonner sur ces énoncés. Parmi les logiques formelles il y a une différence importante entre les logiques classiques et les logiques modales. Dans les logiques classiques la valeur de vérité d'un énoncé est définie par **un** état du monde. Pour savoir, par exemple, si l'énoncé "Jean habite Paris" est vrai ou faux, il suffit de considérer l'état du monde où on se place. Dans les logiques modales, la valeur de vérité des énoncés modaux est définie par **un ensemble** d'états du monde. Par exemple, pour savoir si "Jean a toujours habité à Paris" est vrai ou faux, on doit considérer tous les états du monde que réfère la modalité temporelle "toujours". Pour chaque type de modalité cet ensemble de mondes doit être clairement défini. Ici, par exemple, on pourrait convenir que c'est l'ensemble des états du monde depuis la naissance de Jean jusqu'à l'instant où on se place ; la valeur de vérité peut donc changer en fonction de cet instant.

Plus généralement les logiques modales permettent de distinguer, pour un énoncé p , et un ensemble de mondes donnés :

— le fait que p est vrai dans tous les mondes. Ceci est noté $\Box p$, et on dit que p est

2. Quand on parle d'une représentation du monde il s'agit d'une représentation qui peut en être donnée dans un langage formel fixé. Le choix de ce langage dépend des faits auxquels on s'intéresse dans une application donnée.

“nécessairement vrai”,

— le fait que p est vrai dans au moins un monde. Ceci est noté $\diamond p$, et on dit que p est “possiblement vrai”,

— le fait que p est faux dans au moins un monde. Ceci est noté $\diamond \neg p$, et on dit que p est “possiblement faux”, et

— le fait que p est faux dans tous les mondes. Ceci est noté $\Box \neg p$, et on dit que p est “nécessairement faux”, .

L’ensemble de mondes qui définit le sens d’une modalité peut souvent être interprété comme un contexte. Par exemple, le contexte des connaissances de l’agent Pierre. Alors, ce qui est nécessairement vrai dans ce contexte représente ce que connaît Pierre, et peut être dénoté par l’opérateur modal K_{Pierre} . De la même manière, un autre contexte peut caractériser le contenu d’une base de données, disons “ce que croit la base”, et être dénoté par l’opérateur modal B_{bd} . Enfin, un troisième contexte peut caractériser ce qui est conforme à une réglementation qui définit les autorisations d’accès à une base. Dans ce cas, ce qui est nécessairement vrai dans ce contexte représente ce qui est obligatoire, et ce qui est possiblement vrai ce qui est permis ; qui est dénoté par l’opérateur modal P .

Ces modalités permettent de distinguer, par exemple, le fait que la base bd ne croit pas que Jean habite Paris, noté $\neg B_{\text{bd}}(\text{Jean habite Paris})$, et le fait que la base croit que Jean n’habite pas Paris, noté $B_{\text{bd}}(\neg \text{Jean habite Paris})$. Elles permettent également de représenter le fait que deux bases bd et bd' ont des croyances contradictoires, par exemple avec : $B_{\text{bd}}(\text{Jean habite Paris}) \wedge B_{\text{bd}'}(\neg \text{Jean habite Paris})$. Les opérateurs modaux peuvent aussi être composés entre eux, comme dans l’énoncé : $P(K_{\text{Pierre}}(B_{\text{bd}}(\text{Jean habite Paris} \vee \text{Jean habite Toulouse})))$, qui exprime qu’il est permis que Pierre sache que la base bd croit que Jean habite Paris ou Toulouse.

Comme pour la logique classique, une logique modale est définie par son langage, sa sémantique et/ou son axiomatique. La sémantique définit ce que veut dire le fait qu’un énoncé est vrai dans un monde donné. L’axiomatique définit les règles qui permettent de calculer les conséquences d’un ensemble d’hypothèses. Quand les deux sont définies on cherche habituellement à ce qu’elles se correspondent, dans le sens où l’axiomatique est valide (les énoncés déduits d’hypothèses qui sont vraies sont vrais) et complète (tous les énoncés qui sont vrais, quand certaines hypothèses sont vraies, peuvent être déduits de ces hypothèses).

Nous allons définir formellement la sémantique et l’axiomatique des logiques modales les plus courantes [CHE 88]. Par soucis de simplification on s’est limité au calcul des propositions, pour l’extension au calcul du premier ordre on peut consulter [HUG 72].

2.1. Langage du calcul des propositions modal

On notera par abrégé CPM le langage du calcul des propositions modal.

Alphabet. L'alphabet de CPM est composé d'un ensemble dénombrable de noms de propositions atomiques, a, b, c, \dots ; de deux connecteurs logiques : \neg, \vee et d'un opérateur modal : \Box .

Syntaxe. Les noms des propositions atomiques appartiennent à CPM ; Si p et q sont des formules de CPM, alors $\neg p, p \vee q$, et $\Box p$ sont des formules de CPM ; Il n'y a pas d'autres formules dans CPM que celles définies par les deux règles précédentes.

On appelle CPC (le calcul des propositions classique) le sous-ensemble des formules de CPM qui ne contiennent pas d'opérateurs modaux.

Définitions. On définit d'autres connecteurs logiques de la manière suivante : $p \wedge q \stackrel{\text{def}}{=} \neg(\neg p \vee \neg q)$, $p \rightarrow q \stackrel{\text{def}}{=} \neg p \vee q$, et $p \leftrightarrow q \stackrel{\text{def}}{=} (p \rightarrow q) \wedge (q \rightarrow p)$. L'opérateur modal \Diamond est défini en fonction de l'opérateur \Box par : $\Diamond p \stackrel{\text{def}}{=} \neg \Box \neg p$.

2.2. Sémantique

Il existe deux types de logiques modales : les unes dites normales, et les autres dites classiques. On définit dans un premier temps la sémantique des logiques modales normales.

Logiques modales normales

Une structure M est définie comme un triplet $M = \langle W, R, T \rangle$ où :

- W est un ensemble de mondes possibles.
- R est une relation binaire définie sur $W \times W$.
- T est une fonction qui assigne à chaque nom de proposition atomique de CPM un sous-ensemble de W .

Extension de T . La fonction T est étendue aux formules contenant des connecteurs logiques de la manière suivante.

- $T(\neg p) = W \setminus T(p)$
- $T(p \vee q) = T(p) \cup T(q)$

Conditions de satisfaisabilité. On note $M, w \models p$ le fait que la formule p est vraie dans le monde w de la structure M . Les conditions de satisfaisabilité sont les suivantes.

- $M, w \models p$ ssi w appartient à $T(p)$.
- $M, w \models \Box p$ ssi pour tout monde w' tel que wRw' on a $M, w' \models p$.

D'après la définition de \Diamond , on en déduit :

- $M, w \models \Diamond p$ ssi il existe un monde w' tel que wRw' et $M, w' \models p$.

Si on a plusieurs modalités $\Box_1, \Box_2, \dots, \Box_n$, la sémantique s'étend naturellement à des structures où il y a autant de relations R_i que de modalités \Box_i .

La définition de T montre que les connecteurs logiques \neg et \vee ont leur signification habituelle. Dans un modèle M donné, $T(p)$ définit l'ensemble des mondes où la formule p est vraie. On peut considérer que cet ensemble de mondes caractérise la proposition qui est représentée par la formule p .

Les conditions de satisfaisabilité de $\Box p$ montrent que, dans un structure donnée M ,

la relation R définit, pour chaque monde w , l'ensemble des mondes qui déterminent la valeur de vérité de $\Box p$. La relation R , appelée "relation d'accessibilité", donne son sens à la modalité \Box .

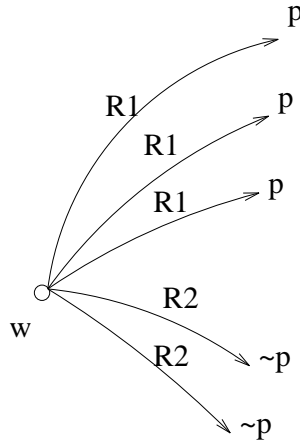


Figure 1. Sémantique de B_{bd1} et B_{bd2}

Par exemple, sur la Figure 1, les modalités B_{bd1} et B_{bd2} sont interprétées par les relations d'accessibilité $R1$ et $R2$. Le fait que $bd1$ (resp. $bd2$) croit p (resp. $\sim p$) est représenté par le fait que dans tous les mondes accessibles depuis w par $R1$ (resp. $R2$) p (resp. $\sim p$) est vraie. Donc en w on a $B_{bd1}(p) \wedge B_{bd2}(\sim p)$. Sur la Figure 2, les modalités P et K_{Pierre} sont interprétées par R et R' . Dans le monde w' on a $K_{Pierre}(p \vee q)$, et dans le monde w on a $P(K_{Pierre}(p \vee q))$, car la modalité P (pour : "permission") représente une possibilité et non une nécessité, et il suffit que $K_{Pierre}(p \vee q)$ soit vrai dans un monde accessible par R .

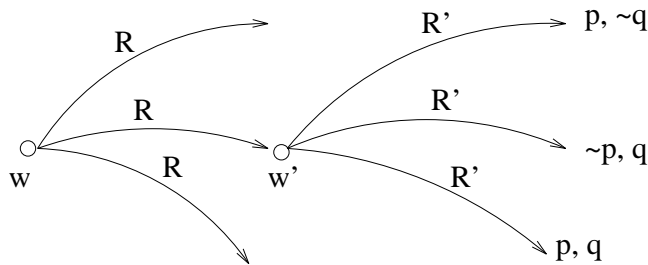


Figure 2. Sémantique de P et K_{Pierre}

Notations. On note $M \models p$ ssi pour tous les mondes w de W on a $M, w \models p$, on dit alors que M est un modèle de p . On note $\models p$ ssi pour toutes les structures on a

$M \models p$, on dit alors que p est valide. D'autre part, une structure M est un modèle d'une logique L ssi c 'est un modèle de tous les théorèmes de cette logique. Pour indiquer que p est vraie dans un modèle M de L on adopte habituellement la notation : $M, w \models_L p$, de même si p est vraie dans tous les modèles de la logique L on a : $\models_L p$. Quand il n'y a pas de risque de confusion on ne mentionne pas L .

Propriétés. Pour les logiques modales normales on a les propriétés suivantes.³

$$\begin{aligned} \text{(C)} \quad & \models \Box p \wedge \Box q \rightarrow \Box p \wedge q \\ \text{(M)} \quad & \models \Box p \wedge q \rightarrow \Box p \wedge \Box q \\ \text{(N)} \quad & \models \Box \top \\ \text{(RE)} \quad & \models p \leftrightarrow q \Rightarrow \models \Box p \leftrightarrow \Box q \end{aligned}$$

Les propriétés C, M et RE impliquent la propriété :

$$\text{(K)} \quad \models \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$$

Dans la logique **KD**, utilisée dans de nombreux domaines, on impose la propriété supplémentaire :

$$\text{(D)} \quad \models \Box p \rightarrow \neg \Box \neg p$$

Les structures sont des modèles de D ssi la relation R est sérielle (tout monde a un correspondant au moins par R). La propriété D peut s'exprimer aussi sous la forme équivalente $\models \Diamond \top$.

Remarque. Quand on impose D on ne peut avoir à la fois $\models \Box p$ et $\models \Box \neg p$, donc on ne peut avoir de contradictions dans le contexte de cette modalité.

Logiques modales classiques

Les logiques modales classiques sont plus faibles que les normales. En particulier, elles ne satisfont pas nécessairement (C), ni (M), ni, a fortiori, (K).

Une structure M est définie comme un triplet $M = \langle W, f, T \rangle$, où W et T sont définis comme pour les logiques normales, et où :

— f est une fonction qui assigne à chaque monde de W un ensemble de sous-ensembles de W .

Conditions de satisfaisabilité.

- $M, w \models p$ ssi w appartient à $T(p)$.
- $M, w \models \Box p$ ssi $T(p)$ appartient à $f(w)$.

Commentaire. Les éléments de $f(w)$ sont des ensembles de mondes qui peuvent (s'ils sont finis) être vus chacun comme représentant une proposition qui peut, elle même, être représentée par une formule. Donc la fonction f fait correspondre à chaque monde un ensemble de propositions (voir Figure 3). Si on veut représenter plusieurs modalités, on définit dans les structures autant de fonctions f .

Propriété. Pour les logiques modales classiques on a la propriété suivante.

$$\text{(RE)} \quad \models p \leftrightarrow q \Rightarrow \models \Box p \leftrightarrow \Box q$$

On peut enrichir une logique modale classique en imposant des contraintes aux modèles de ces logiques. Par exemple, on aura la propriété C ssi les modèles satisfont la propriété suivante :

3. On note \top une tautologie quelconque, et \perp une contradiction quelconque.

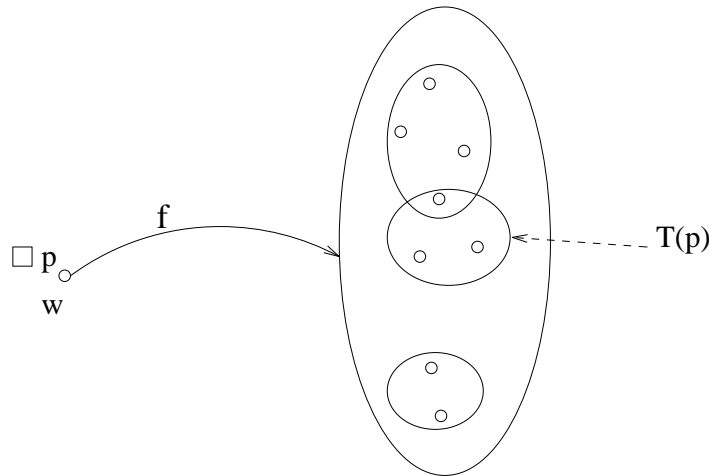


Figure 3. Sémantique de $\Box p$ dans les logiques modales classiques

si $X \in f(w)$ et $Y \in f(w)$ alors $X \cap Y \in f(w)$

On peut enrichir les logiques modales classiques jusqu'à avoir des logiques modales normales. D'autre part on peut représenter une logique modale normale comme un cas particulier de logique modale classique qui est définie ainsi. A chaque relation d'accessibilité R de la logique modale normale il suffit de faire correspondre, dans la logique modale classique, une fonction f définie par $f(w) = \{X(w)\}$, avec $X(w) = \{w' : wRw'\}$.

Remarque. Dans un monde w on peut avoir $\emptyset \in f(w)$, ce qui signifie que \perp est vraie dans ce monde. On peut aussi avoir $f(w) = \emptyset$, ce qui signifie qu'il n'y a aucune formule p telle que $\Box p$ soit vraie en w .

2.3. Axiomatique

Les axiomatiques des logiques modales normales et classiques sont définies ici⁴ par des systèmes de Hilbert.

Logiques modales classiques

- schémas d'axiomes du calcul des propositions classique.
- règles d'inférence :
 - modus ponens : (MP) $\frac{\vdash p, \vdash p \rightarrow q}{\vdash q}$
 - substitutivité des équivalents : (RE) $\frac{\vdash p \leftrightarrow q}{\vdash \Box p \leftrightarrow \Box q}$

4. On peut également définir l'axiomatique par des systèmes à la Gentzen, mais les systèmes de Hilbert nous paraissent plus appropriées pour une compréhension intuitive.

Logiques modales normales

En plus de l'axiomatique des logiques modales classiques, on a les schémas d'axiomes suivants.

$$\begin{array}{ll}
 \text{(C)} & \Box p \wedge \Box q \rightarrow \Box p \wedge q \\
 \text{(M)} & \Box p \wedge q \rightarrow \Box p \wedge \Box q \\
 \text{(N)} & \Box \top
 \end{array}$$

Pour les logiques modales normales on peut choisir indifféremment les systèmes formés avec (C), (M), (N) et (RE), ou bien avec (K) et la règle d'inférence : $(\text{Nec}) \frac{\vdash p}{\vdash \Box p}$. On obtient la logique **KD** en rajoutant le schéma d'axiome : $(D) \vdash \Box p \rightarrow \neg \Box \neg p$.

3. Fiabilité des sources d'informations**3.1. Présentation du problème**

Les informations qui sont insérées dans une base de données ne peuvent jamais être toutes garanties comme étant correctes vis-à-vis de la réalité qu'elles représentent, même si elles sont cohérentes avec les contraintes d'intégrité. Néanmoins, d'un point de vue pragmatique, on peut considérer, au moins pour certaines d'entre elles, qu'elles sont correctes. Le fait qu'elles soient ou non, garanties correctes dépend, en dernière analyse, de la fiabilité des sources d'informations qui sont à l'origine des insertions [DEM 94b].

Par exemple, on peut accepter, dans une entreprise, que si une information sur le salaire d'un employé est insérée par un agent du service comptabilité, alors cette information est vraie. Dans ce cas on dira que cet agent est fiable pour la validité des informations sur les salaires, dans le sens où, s'il affirme, (c'est-à-dire, s'il insère dans la base) que tel employé a tel salaire, alors cela est vrai. De façon duale, on peut dire qu'un agent du service gestion du personnel est fiable pour la complétude des informations concernant l'absentéisme des employés, dans le sens où, si un employé vient à être absent alors ce fait est inséré par cet agent dans la base.

Quand les informations ont été insérées dans la base par de nombreux agents, qui sont, ou ne sont pas, fiables pour la validité ou pour la complétude de tel ou tel type d'information, alors il n'est pas facile de déterminer la validité ou la complétude de la réponse à une question complexe posée à cette base. Pour analyser formellement les concepts de fiabilité vis-à-vis de la validité ou de la complétude, on a défini une logique appropriée. Cette logique nous a permis de caractériser les parties d'une réponse qui sont garanties valides et celles qui sont garanties complètes [CHO 94d, DEM 96b, DEM 96a].

3.2. Modélisation en logique

Dans cette logique le fait que l'information p fait partie (explicitement, ou implicitement, par déduction) de la base est représenté par Bp . On peut lire Bp : "la base de données croit que p est vrai". Le fait que l'information p fasse partie de la base, et qu'elle soit garantie comme vraie (ou du moins réputée telle), est représenté par Kp . On peut lire Kp : "la base de données "sait" que p est vraie". Intuitivement, si la base "sait" que p , alors la vérité de p , du point de vue de la base, ne peut être mise en question. C'est-à-dire que, non seulement la base croit p , mais elle croit que c'est une croyance exacte. Ces propriétés de la modalité K sont formalisées dans la logique par les schémas d'axiomes :

$$\begin{array}{ll} \text{(KB)} & Kp \rightarrow Bp \\ \text{(T')} & K(Kp \rightarrow p) \end{array}$$

On suppose la base de données cohérente, donc on accepte pour les modalités K et B les propriétés de la logique **KD**.

La validité des informations présentes dans la base dépend des agents qui les ont insérées. Donc, pour raisonner sur leur validité, il faut pouvoir représenter les actes d'insertion. En fait, on n'a pas besoin de représenter le détail des opérations qui conduisent à une insertion, mais ce qu'il faut représenter c'est le lien entre le résultat de l'action, par exemple Bp , et l'agent i qui est à l'origine, qui est la cause, du fait que l'on a Bp .

Ce lien peut être exprimé par un opérateur d'action, qui a été largement étudié dans la littérature [POR 77, CHE 95], qu'on appelle l'opérateur de "faire en sorte que". Il est noté E_i , et le fait $E_i(Bp)$ peut se lire: "l'agent i a fait en sorte que la base croit p ". Il est important que cet opérateur E_i exprime effectivement la notion de causalité. En effet, si, par exemple, un agent i du service comptabilité donne l'ordre à une secrétaire j d'insérer dans la base le fait que telle personne a tel salaire, on ne peut pas dire que c'est la secrétaire qui est la cause de l'insertion. Ce n'est pas elle qui a fait le choix d'insérer p , et donc on ne peut pas dire que c'est elle qui a fait en sorte que l'on ait Bp car, en réalité, c'est l'agent i , car c'est lui qui a fait le choix. Cette situation sera donc représentée par $E_i(Bp)$ et non par $E_j(Bp)$. Le fait que le salaire soit exact dépend de la fiabilité, vis-à-vis de la validité, de l'agent i qui est à l'origine de l'insertion, et non de la fiabilité de la secrétaire.

La logique que nous avons retenue pour formaliser l'opérateur de "faire en sorte que" est plus faible que les systèmes normaux. C'est un système classique qui satisfait (RE) et (\neg N) :

$$\begin{array}{ll} \text{(RE)} & \frac{\vdash p \leftrightarrow q}{\vdash E_i p \leftrightarrow E_i q} \\ (\neg\text{N}) & \neg E_i \top \end{array}$$

Le schéma (\neg N) exprime, en partie, qu'un agent ne peut pas être la cause de ce qu'un fait soit vrai, si la vérité de ce fait, ici n'importe quelle tautologie, ne dépend pas de lui. Les modalités K , B et E permettent de donner une définition précise de la notion de fiabilité des sources d'information.

On dit qu'un agent i est fiable pour la validité de l'information p ssi la base "sait" que si cet agent fait en sorte que la base croit p , alors p est vraie. Cette propriété est notée $RV_i(p)$, et on a⁵ :

$$RV_i(p) \stackrel{\text{def}}{=} K(E_i(Bp) \rightarrow p)$$

De façon similaire, on dit qu'un agent i est fiable pour la complétude de p ssi la base "sait" que si p est vrai, alors cet agent a fait en sorte que la base croit p . Cette propriété est notée $RC_i(p)$, et on a :

$$RC_i(p) \stackrel{\text{def}}{=} K(p \rightarrow E_i(Bp))$$

Ces définitions se généralisent⁶ facilement aux énoncés de la forme $p(x)$; on a⁷ :

$$RV_i(p(x)) \stackrel{\text{def}}{=} K(\forall x(E_i(Bp(x)) \rightarrow p(x)))$$

$$RC_i(p(x)) \stackrel{\text{def}}{=} K(\forall x(p(x) \rightarrow E_i(Bp(x))))$$

Dans la pratique ce sont généralement ces définitions qui seront utilisées car on sait, par exemple, qu'un agent est fiable pour la validité des salaires en général, et pas simplement pour tel salaire particulier. Ce qui sera exprimé par $RV_i(\text{salaire}(x, y))$.

On notera que les deux définitions de la fiabilité ont une forme conditionnelle. C'est parce qu'elles définissent des propriétés des agents qui sont indépendantes des insertions qu'ils ont, ou n'ont pas, réalisées à un moment donné. Pour une application, le fait d'accepter l'hypothèse qu'un agent i est fiable pour la validité de p (formellement : $RV_i(p)$) ou la complétude de p (formellement : $RC_i(p)$), ne dépend pas du fait que l'agent i a, ou n'a pas, inséré l'information p (formellement : $E_i(Bp)$ ou $\neg E_i(Bp)$).

Enfin, on se donne dans cette logique les schémas d'axiome (OBS1) et (OBS2), qui expriment que la base a connaissance de toutes les insertions qui sont réalisées :

$$(OBS1) \quad E_i(Bp) \rightarrow K(E_i(Bp))$$

$$(OBS2) \quad \neg E_i(Bp) \rightarrow K(\neg E_i(Bp))$$

et le schéma (B) qui exprime que la base est crédule, dans le sens où, si un agent a fait en sorte qu'elle croit p , elle accepte cette croyance :

$$(B) \quad K(E_i(Bp)) \rightarrow Bp$$

3.3. Un exemple d'application

Considérons un exemple d'école. Soit, dans une université, un agent i qui est fiable pour la validité des informations exprimant que des étudiants sont inscrits à un module

5. Nous avons gardé les notations anglaises où RV est une abréviation pour "reliable for validity", et RC pour "reliable for completeness".

6. On suppose que les variables quantifiées sont rigides, c'est-à-dire sont interprétées par le même individu dans tous les mondes.

7. Pour simplifier les notations nous avons considéré une seule variable libre mais il peut y en avoir plusieurs.

(représentées par le prédicat : $\text{inscrit}(x)$), et un agent j qui est fiable pour la complétude des informations exprimant que des étudiants ont été reçus à l'examen de ce module (représentées par le prédicat : $\text{reçu}(x)$).

Supposons que l'agent i ait inséré le fait que Pierre et Paul sont inscrits, et que l'agent j ait uniquement inséré le fait que Pierre est reçu. Cette situation est formellement décrite par les ensembles de faits bd et mbd , qui décrivent respectivement le contenu de la base, et les méta informations concernant la fiabilité des agents.

$$\begin{aligned} \text{bd} &= \{E_i(\text{Binscrit}(\text{Pierre})), E_i(\text{Binscrit}(\text{Paul})), \\ &\quad E_j(\text{Breçu}(\text{Pierre})), \neg E_j(\text{Breçu}(\text{Paul}))\} \\ \text{mbd} &= \{\text{RV}_i(\text{inscrit}(x)), \text{RC}_j(\text{reçu}(x))\} \end{aligned}$$

Si maintenant on veut connaître l'ensemble des étudiants dont on peut garantir qu'ils sont inscrits et qu'ils ne sont pas reçus, on doit, formellement, rechercher l'ensemble des x tels que : $\vdash \text{bd} \wedge \text{mbd} \rightarrow K(\text{inscrit}(x) \wedge \neg \text{reçu}(x))$.

Pour cela on peut faire la déduction suivante. De bd , avec (OBS1), on déduit $K(E_i(\text{Binscrit}(\text{Paul})))$, et, d'après mbd , on a : $K(E_i(\text{Binscrit}(\text{Paul})) \rightarrow \text{inscrit}(\text{Paul}))$. Comme la modalité K est normale, on déduit : $K\text{inscrit}(\text{Paul})$.

D'autre part, de bd , avec (OBS2), on déduit $K(\neg E_j(\text{reçu}(\text{Paul})))$. D'après mbd on a : $K(\text{reçu}(\text{Paul}) \rightarrow E_j(\text{Breçu}(\text{Paul})))$, et comme K satisfait (RE), on a : $K(\neg E_j(\text{Breçu}(\text{Paul})) \rightarrow \neg \text{reçu}(\text{Paul}))$, d'où : $K(\neg \text{reçu}(\text{Paul}))$. Donc, finalement on a :

$K(\text{inscrit}(\text{Paul}) \wedge \neg \text{reçu}(\text{Paul}))$. On notera que ce qui permet de garantir que Paul n'est pas reçu, c'est la fiabilité de j pour la complétude, et non pour la validité, comme on pourrait le croire au premier abord. Intuitivement, ce qui garanti que Paul n'est pas reçu, c'est le fait que l'agent j n'a oublié aucun des reçus.

Si, d'autre part, on s'intéresse aux étudiants qui sont inscrits et reçus, on remarque qu'on ne peut pas garantir que Pierre répond à cette question, car on a : $\vdash \text{bd} \wedge \text{mbd} \rightarrow K\text{inscrit}(\text{Pierre})$, mais on n'a pas : $\vdash \text{bd} \wedge \text{mbd} \rightarrow K\text{reçu}(\text{Pierre})$. Par contre, de bd , avec (OBS1) et (B), on peut déduire $\text{Binscrit}(\text{Pierre})$ et $\text{Breçu}(\text{Pierre})$, d'où : $\vdash \text{bd} \wedge \text{mbd} \rightarrow B(\text{inscrit}(\text{Pierre}) \wedge \text{reçu}(\text{Pierre}))$. La base croit que Pierre est reçu, mais cela n'est pas garanti.

Cette logique a été particularisée pour les bases de données relationnelles pour lesquelles on accepte la CWA (en français : Hypothèse du Monde fermé). Les axiomes de complétion de la CWA sont exprimés ici à l'aide des schémas d'axiomes complémentaires suivants :

$$\begin{aligned} (1) \quad & B(p \vee q) \rightarrow Bp \vee Bq \\ (2) \quad & \neg Bp \rightarrow B(\neg p) \end{aligned}$$

Dans ce contexte particulier, on a transposé la technique de déduction du langage du calcul des prédicats au langage de l'algèbre relationnelle [DEM 96b] , ce qui permet d'envisager une implémentation plus efficace, et compatible avec les systèmes relationnels existants.

4. Fusion de données

4.1. Présentation du problème

De plus en plus, les applications informatiques ont besoin d'utiliser des informations qui ne sont pas fournies par une seule source, mais par plusieurs. C'est le cas, par exemple, des bases de données fédérées. Chaque base de données stocke des données relatives à un domaine d'application particulier et lorsque l'on considère un domaine plus vaste, il faut alors fédérer différentes bases. En général, la fusion des bases de données est virtuelle, dans le sens où à aucun moment on ne construit réellement une nouvelle base de données. Un autre exemple d'applications où la fusion de données est un problème central, se trouve dans le contexte des SIC (Systèmes d'Informations et de Communication) de renseignements militaires, qui sont des systèmes chargés, entre autres, de centraliser des informations provenant de sources différentes, afin que des officiers puissent prendre des décisions. Si, jusqu'à présent la fonction de fusion était assurée par des officiers de renseignements, il devient souhaitable qu'elle soit assistée par des outils informatiques.

L'exemple que nous prendrons dans ce paragraphe, pour illustrer nos propos, est celui d'un inspecteur de police qui collecte les différents témoignages relatifs à un accident de la route: plusieurs témoins disent ce qu'ils ont vu, ou plutôt ce qu'ils pensent avoir vu, et l'inspecteur doit fusionner ces informations afin de se faire une idée de la réalité. Cet exemple illustre à la fois le cas des bases de données fédérées et le cas des SIC, dans la mesure où, comme eux, il pose le problème de la fusion de croyances émises par des sources plus ou moins crédibles. En effet, dans certains cas, l'inspecteur peut supposer que tel témoin est crédible et qu'il dit la vérité. Si l'on reprend le formalisme décrit au paragraphe précédent, le fait qu'un agent est crédible est exprimé par le fait qu'il est fiable pour la validité. Soit formellement : $K_{\text{inspecteur}}(E_{\text{témoin}}B_{\text{inspecteur}}P \rightarrow P)$.

Par contre, dans d'autres cas, l'inspecteur a uniquement une idée de la crédibilité relative des témoins, et peut seulement supposer que tel témoin est plus crédible que tel autre.

Le problème de la modélisation de la fiabilité des sources pour la validité des informations, a été abordé dans le paragraphe précédent et nous n'y reviendrons pas ici. Dans ce paragraphe, nous nous focaliserons sur le problème de la fusion de données contradictoires. En effet, si l'on ne peut pas garantir que toutes les sources sont crédibles, il peut se faire que des sources émettent des données contradictoires, même si, indépendamment les unes des autres, les sources sont cohérentes.

Dans le contexte de la fusion de renseignements, on retrouve ce problème du fait de la grande diversité des sources d'informations et des techniques qu'elles utilisent pour observer une situation. L'inspecteur de police lui-même se trouve souvent face à des témoignages contradictoires, notamment si l'un des témoins est lui-même complice et a intérêt à mentir.

4.2. Modélisation en logique

Dans le contexte de la fusion de données contradictoires, nous avons montré qu'ordonner les différentes sources d'informations à fusionner était une bonne hypothèse de travail : en effet, cet ordre peut refléter le fait que, lors de la fusion, les sources sont considérées comme plus ou moins crédibles. L'idée intuitive étant qu'on gardera de préférence les informations provenant d'une source plus crédible. En d'autres termes, on *croira plus* ce que dit une source plus crédible.

Prenons à nouveau l'exemple de l'inspecteur de police. Supposons que le premier témoin, Pierre, ait dit qu'il avait vu, sur le lieu de l'accident, une voiture de sport, tandis que le deuxième témoin, Jean, a dit qu'il avait vu une femme dans une voiture de type diesel. L'inspecteur, après s'être renseigné à la station météorologique, est certain que l'accident s'est déroulé dans un épais brouillard. Or Jean était situé assez loin du lieu de l'accident, tandis que Pierre lui, en était plus proche. L'inspecteur peut donc supposer que le témoignage de Jean est moins fiable que celui de Pierre. Faire cette hypothèse revient, pour l'inspecteur, à considérer l'ordre suivant : station-météo > Pierre > Jean.

Cependant, un tel ordre entre les témoins peut être utilisé lors de la fusion de deux façons différentes [CHO 92], [CHO 94b] :

— Jean contredit Pierre or Jean est moins crédible que Pierre. Donc, l'inspecteur, s'il est suspicieux, ne va rien retenir du témoignage de Jean.

Cette attitude, que l'on appelle "suspicieuse" consiste à rejeter toute information provenant d'une source qui contredit, pour une information au moins, une source supposée plus crédible qu'elle même.

— Jean est moins crédible, Jean contredit Pierre à propos du type de la voiture (en effet sport et diesel sont contradictoires). Donc l'inspecteur retiendra l'information fournie par Pierre sur le type de la voiture, à savoir que c'était une voiture de sport. Par contre, Jean apporte une information sur la personne qui occupait la voiture. Or Pierre ne dit rien là dessus. Donc, l'inspecteur, s'il est moins suspicieux que précédemment, retiendra le fait qu'il y avait une femme dans la voiture.

Cette attitude, que l'on appelle "confiante" consiste, en cas de contradiction, à ne rejeter que les informations contradictoires les moins crédibles.

Chacune des deux attitudes précédentes a été étudiée. Elles ont conduit à deux logiques de la fusion des croyances. Dans ce qui suit, nous ne présentons que FUSION-C, la logique correspondant à l'attitude confiante.

Notons $1, \dots, n$ les sources d'informations à fusionner. On suppose dans un premier temps, que ce sont des ensembles de littéraux positifs ou négatifs, consistants.

Dans cette approche, fusionner k sources d'informations, par exemple les k premières, en supposant que 1 est plus crédible que 2, 2 plus crédible que 3, ..., $k - 1$ plus crédible que k , revient à construire une nouvelle source d'informations, que l'on notera $1 > \dots > k$.

Dans la logique FUSION-C, définie pour simuler la fusion de données selon une attitude confiante, le fait qu'une information p soit fournie par la source d'informations

i est représenté par la formule $B_i p$, qui se lit “la source i croit que p est vraie”. Le fait qu’une source $1 > \dots > k$, obtenue par fusion, croit l’information p est représenté par la formule $B_{1>\dots>k} p$.

Les modalités de la forme B_i et les modalités de la forme $B_{1>\dots>k}$ sont régies par les axiomes de la logique **KD**. En d’autres termes, les croyances d’une source d’informations (qu’elle soit primitive ou obtenue par fusion) sont cohérentes et fermées pour la déduction.

De plus, parce que l’on se restreint à des ensembles de littéraux, on a l’axiome suivant :

$$— (0) B_o (l_1 \vee \dots \vee l_m) \rightarrow B_o l_1 \vee \dots \vee B_o l_m$$

Cet axiome, où “ o ” représente un ordre total entre des sources (éventuellement réduit à un singleton) signifie qu’une base (primitive ou obtenue par fusion) ne croit une assertion disjonctive que si elle croit l’un des éléments de la disjonction.

Les liens entre les modalités de la forme B_i et les modalités de la forme $B_{1>\dots>k}$ sont définis par les axiomes suivants, où l est un littéral de L :

$$— (1) B_o l \rightarrow B_{o>i} l$$

$$— (2) B_i l \wedge \neg B_o \neg l \rightarrow B_{o>i} l$$

$$— (3) B_{o>i} l \rightarrow B_o l \vee B_i l$$

Ces axiomes, où “ o ” est un ordre total entre des sources (éventuellement réduit à un singleton) et i est une source, formalisent l’attitude confiante de la fusion.

En effet, selon (1), tout fait fourni par une base (primitive ou obtenue par fusion) appartient à la base obtenue en fusionnant cette base avec une autre base considérée moins crédible. En d’autres termes, lors de la fusion, on retient tous les faits fournis par une base plus crédible.

Selon (2) tout fait fourni par une base (primitive) appartient à la base obtenue en fusionnant cette base avec une autre considérée comme plus crédible, à condition que cette dernière ne croit pas le contraire. En d’autres termes, lors de la fusion, on retient tous les faits fournis par une base la moins crédible, à condition qu’ils ne soient pas en contradiction avec des faits fournis par la base la plus crédible.

(3) exprime que, l’acte de fusionner ne rajoute pas d’informations. Tous les faits qui apparaissent dans une base résultant d’une fusion, sont fournis par l’une ou l’autre des bases à fusionner.

4.3. Un exemple d’application

Illustrons cette logique sur l’exemple suivant. Un inspecteur de police interroge les deux témoins d’un accident. Le premier témoin, Pierre, dit qu’il avait vu une voiture de sport conduite par une femme habillée d’une robe. Le deuxième témoin, Jean, dit qu’il avait vu une voiture diesel conduite par une femme habillée d’un tailleur.

$$\text{Pierre} = \{\text{sport}, \neg \text{diesel}, \text{robe}, \neg \text{tailleur}\}.$$

$$\text{Jean} = \{\neg \text{sport}, \text{diesel}, \text{tailleur}, \neg \text{robe}\}$$

Notons ψ , l'ensemble qui énumère ce que croient, et ce que ne croient pas, Jean et Pierre.

$$\psi = \{ B_{\text{Pierre}}\text{sport}, B_{\text{Pierre}}\neg\text{diésel}, B_{\text{Pierre}}\text{robe}, B_{\text{Pierre}}\neg\text{tailleur}, \dots \\ B_{\text{Jean}}\text{diésel}, B_{\text{Jean}}\neg\text{sport}, B_{\text{Jean}}\neg\text{robe}, B_{\text{Jean}}\text{tailleur}, \dots \}.$$

Notons $\vdash_{\text{FC}} F$, le fait que F soit un théorème de la logique précédente. Voici quelques conclusions que peut tirer l'inspecteur selon une attitude confiante :

$$\vdash_{\text{FC}} \psi \rightarrow B_{\text{Pierre} > \text{Jean}}(\text{robe} \wedge \text{sport} \wedge \neg\text{diésel} \wedge \neg\text{tailleur})$$

Si l'inspecteur considère que Pierre est plus crédible que Jean, alors il va déduire qu'il y avait une femme, habillé d'une robe et non en tailleur, dans une voiture de sport et non une voiture diésel.

$$\vdash_{\text{FC}} \psi \rightarrow B_{\text{Jean} > \text{Pierre}}(\text{tailleur} \wedge \neg\text{robe} \wedge \neg\text{sport} \wedge \text{diésel})$$

Si l'inspecteur considère que Jean est plus crédible que Pierre, alors il va déduire qu'il y avait une femme habillée en tailleur et non en robe dans une voiture diésel, et non une voiture de sport.

$$\vdash_{\text{FC}} \psi \rightarrow \neg B_{\text{Jean} > \text{Pierre}}\text{sport}$$

Si l'inspecteur considère que Jean est plus crédible que Pierre, il ne va pas déduire que la voiture était une voiture de sport. C'est normal puisque dans ce cas, il déduit qu'elle n'était pas une voiture de sport.

4.3.1. Extensions

Bases de données avec règles

Dans [CHO 96], nous présentons une extension de FUSION-C qui permet de considérer des sources d'informations qui partagent un ensemble commun de règles (règles de déduction, contraintes d'intégrité). La notion de contradiction entre deux sources est légèrement différente et tient compte de ces règles. Ainsi, intuitivement, deux sources sont contradictoires s'il existe des littéraux de l'une et des littéraux de l'autre, qui sont contradictoires avec les règles.

Cette extension nous permettrait de reprendre l'exemple ci-dessus, en considérant que les témoignages se réduisent aux ensembles : Pierre = {sport, robe} et Jean = {diésel, tailleur} et en considérant les règles : IDB = { $\neg\text{sport} \vee \neg\text{diésel}, \neg\text{robe} \vee \neg\text{tailleur}$ }.

Relation d'ordre dépendante des thèmes

Ce travail a été étendu en considérant que, lors de la fusion, il est en fait difficile de dire si une source est globalement plus crédible qu'une autre. Pour être plus précis, il faut regarder de quoi parlent ces sources, c'est à dire, quels sont les thèmes des informations qu'elles transmettent. La crédibilité relative des sources peut alors être définie thème par thème.

Par exemple, Jean peut être considéré par l'inspecteur comme étant plus crédible que Pierre, en ce qui concerne toutes les informations relatives à la voiture. Par contre, Pierre peut être supposé plus crédible que Jean, en ce qui concerne les informations relatives aux vêtements de la femme qui conduisait.

La logique FUSION-C a été étendue dans le cas où la relation d'ordre entre les bases dépend des thèmes. [CHO 94a], [CHO 94c].

5. Confidentialité dans les bases de données

5.1. Présentation du problème

Une autre application intéressante pour les travaux de modélisation en logique est la sécurisation multi-niveaux de systèmes de gestion de base de données (SGBD). Pour définir une politique de sécurité multi-niveaux, il faut commencer par considérer un ensemble Niveau de niveaux de sécurité, par exemple, Niveau = {Secret, Confidentiel, Public}. On suppose, en général, que l'ensemble Niveau est un treillis associé à un ordre partiel noté $<$. Ainsi, dans notre exemple, on aura Public $<$ Confidentiel $<$ Secret. Dans une politique de sécurité multi-niveaux, des habilitations sont ensuite attribuées aux personnes en fonction du rôle qu'elles jouent dans l'organisation, et une classification est associée aux informations. Dans le cas de la confidentialité, la politique de sécurité multi-niveaux peut se résumer par la règle suivante :

Une personne peut apprendre une information seulement si son niveau d'habilitation domine le niveau de classification de cette information

Denning dans [DEN 82] montre que, pour définir un modèle complet des contraintes que la politique de sécurité multi-niveaux impose au SGBD, il est commode de décomposer le problème de la confidentialité en deux sous-problèmes :

1. Le contrôle des flux d'informations internes au SGBD. Trouver une solution à ce problème fournit l'assurance qu'un utilisateur ne peut obtenir du SGBD que des informations autorisées vis-à-vis de la politique de sécurité multi-niveaux.

2. Le contrôle de l'inférence. Trouver une solution à ce problème fournit l'assurance qu'un utilisateur ne peut pas déduire d'informations interdites vis-à-vis de la politique de sécurité multi-niveaux, en utilisant des informations auxquelles cet utilisateur a légalement accès.

L'objectif de cette section est de montrer comment un formalisme logique fondé sur les concepts de connaissances (logique épistémique), de croyance (logique doxastique) ainsi que de permission et d'interdiction (logique déontique) permet de formaliser les différents concepts intervenant dans une base de données multi-niveaux et d'exprimer divers types de contraintes de sécurité.

5.2. Modélisation en logique

5.2.1. Concept de base de données multi-niveaux

Dans [CUP 96a], nous avons proposé une représentation d'une base de données multi-niveaux. Cette représentation est très simple : il s'agit simplement d'une collection de bases de données. Cela signifie que la base de données multi-niveaux globale est partitionnée en bases de données mono-niveau associées à chaque niveau de sécurité de l'ensemble Niveau. Cette définition est formalisée en logique de la façon suivante.

On suppose que le contenu de la base de données est représenté par un ensemble

de formules d'un langage L du calcul des propositions classique.

Une base de données multi-niveaux BD est alors représentée par un ensemble de bases de données $\{BD_i, i \in \text{Niveau}\}$. A chaque base de données mono-niveau BD_i est associé un ensemble consistant d'informations bd_i .

Dans la suite, on ne fait pas l'hypothèse que toutes les informations contenues dans chaque ensemble bd_i sont vraies dans le monde. On considère donc que chaque bd_i est un ensemble de croyances. Dans un formalisme logique, cela est représenté en introduisant un ensemble de modalités B_i , une formule de la forme $B_i p$ étant lue : la base de données de niveau i croit que p est vrai. Nous supposons que la logique associée à chaque modalité B_i est de type **KD**.

L'étape suivante consiste à construire la vue de la base de données pour chaque niveau de sécurité $i \in \text{Niveau}$. Cette vue est obtenue en fusionnant toutes les bases de données BD_l telles que $l \leq i$. Pour cela, nous avons proposé dans [CHO 95b] d'appliquer la logique FUSION-C, présentée dans la section 4.2, en utilisant l'ordre défini sur les niveaux de sécurité comme ordre de préférence pour fusionner les différentes bases de données BD_l . Cette solution permet de prendre en compte les bases de données multi-niveaux qui utilisent des leurres. Un leurre est une donnée fautive qui est délibérément fournie en réponses à certaines requêtes posées par des personnes non habilitées à connaître l'existence d'une information secrète. Dans ce cas, l'idée intuitive est la suivante : s'il existe une contradiction entre deux bases de données BD_{l_1} et BD_{l_2} ($l_1 < l_2$), alors c'est l'information contenue dans la base de données BD_{l_2} qui sera retenue dans la vue $Vu_{e_{l_2}}$, l'information contradictoire contenue dans la base de données BD_{l_1} étant interprétée comme un leurre. La logique associée aux différentes modalités Vu_{e_i} est également de type **KD**.

Pour étudier les problèmes de confidentialité dans une base de données multi-niveaux, nous avons également besoin de représenter les interactions que peuvent avoir les utilisateurs avec la base de données multi-niveaux. Pour cela, nous introduisons un ensemble de modalités KB_u pour chaque utilisateur u de la base de données multi-niveaux. Une formule de la forme $KB_u p$ doit être lue : l'utilisateur u sait que p est cru par la base de données multi-niveaux⁸. Chaque modalité KB_u correspond donc à la combinaison d'une modalité épistémique de type **KT**⁹ et d'une modalité doxastique de type **KD**. Il est alors facile de vérifier que les modalités KB_u sont donc associées à une logique de type **KD**.

5.2.2. Concept de politique de sécurité

Pour décrire la politique de sécurité multi-niveaux, nous introduisons une fonction qui associe un niveau de classification aux formules du langage L utilisé pour décrire le contenu de la base de données multi-niveaux. Pour représenter cette fonction de classification dans le langage, nous introduisons, pour chaque niveau de sécurité l , une modalité $[l]$, la formule $[l]p$ devant être lue : la formule p est classifiée au niveau

8. En fait, nous supposons que tout utilisateur u interagit avec la vue correspondant à son niveau d'habilitation. Il serait donc plus correct de lire $KB_{u,p}$ de la façon suivante : l'utilisateur u sait que p est cru par la vue de la base correspondant à son niveau d'habilitation.

9. La logique **KT** est une logique modale normale pour laquelle on a les axiomes (K) et (T) ($\Box p \rightarrow p$)

l. La logique associée à cette modalité est une logique modale classique pour laquelle on n'a que la règle d'inférence (RE). Par exemple, l'expression :

— [Secret]Salaire(Dupont, 15000)

exprime le fait que le niveau de classification de la formule Dupont a pour salaire 15000 est égal à secret.

Remarquons que la fonction de classification n'est pas forcément complète : certaines formules de L peuvent ne pas être classifiées. D'autre part, nous supposons que la fonction de classification peut être définie indépendamment du contenu de la base de données multi-niveaux :

1. Une formule peut être classifiée sans pour autant être crue par la base de données. Cette possibilité permet de prendre en compte des règles générales de classification comme par exemple : le salaire de Dupont est secret (et cela indépendamment de la valeur effective du salaire de Dupont). Dans ce cas, toutes les formules de la forme Salaire(Dupont, sal_i), où sal_i est une valeur possible de salaire, seront classées secrètes. Mais, si l'on suppose que le salaire de Dupont est unique et par exemple égal à 15000, seule la formule Salaire(Dupont, 15000) sera effectivement crue par la base de données.

2. Inversement, une formule peut être crue par la base de données BD_i sans pour autant être classifiée au niveau i . Ce genre de situation n'est pas écartée a priori. Ce sera le rôle des contraintes de sécurité (voir section 5.3) de décider si la situation considérée est acceptable ou non.

Pour de simples raisons de commodité de formalisation, nous avons également introduit dans le langage les modalités $[\leq l]$ et $[> l]$ (pour chaque niveau de sécurité l). Ces modalités peuvent s'appliquer à des ensembles quelconques, mais finis, de formules de L [CUP 96b]. Intuitivement, $[\leq l](p_1, \dots, p_n)$ (resp. $[> l](p_1, \dots, p_n)$) doit être lu : p_1, \dots, p_n représente l'ensemble de toutes les formules ayant un niveau de classification inférieur ou égal (resp. strictement supérieur) à l .

A partir de cette notion de classification, nous dérivons les notions déontiques de permission et interdiction. Ces deux notions seront représentées à l'aide des modalités PKB_l et FKB_l . Le fait qu'un utilisateur de niveau d'habilitation l a la permission de savoir que la base de données croit p est représenté par $PKB_l p$, et le fait qu'il est interdit à un utilisateur de niveau d'habilitation l de savoir que la base de données croit que p est représenté par $FKB_l p$.

Par abus de langage $PKB_l p$ et $FKB_l p$ seront par la suite explicités sous la forme "un utilisateur de niveau d'habilitation l a la permission de connaître p " et "il est interdit à un utilisateur de niveau d'habilitation l de connaître p ", étant entendu que "un utilisateur de niveau d'habilitation l connaît p " est un raccourci pour "un utilisateur de niveau d'habilitation l sait que la base de données croit p ". L'axiomatique des modalités PKB_l et FKB_l est définie de la façon suivante¹⁰ :

— Pour la modalité $FKB_l(p)$, nous avons :

10. Dans ces deux axiomes, si $n = 0$, alors $[> l](l_1, \dots, l_n)$ doit être interprété comme $[> l](\perp)$ et $[\leq l](l_1, \dots, l_n)$ doit être interprété comme $[\leq l](\top)$.

$$(F) \quad [>](p_1, \dots, p_n) \rightarrow \\ \left(\bigvee_{i \in [1, n]} \Box(p \rightarrow p_i) \right) \leftrightarrow \text{FKB}_l(p)$$

où la modalité \Box est une modalité de type **KT**, les formules de la forme $\Box p$ devant être lues : p est une tautologie du calcul des propositions.

L'axiome ci-dessus exprime donc qu'un utilisateur de niveau d'habilitation l à l'interdiction de connaître toute formule p qui permet de dériver une formule p_i , le niveau de classification de p_i étant strictement supérieur à l . En particulier, comme on a $\Box(p_i \rightarrow p_i)$, il est facile de vérifier qu'un utilisateur de niveau d'habilitation l a l'interdiction de connaître toute formule p_i ayant un niveau de classification strictement supérieur à l .

— Pour la modalité $\text{PKB}_l(p)$, nous avons :

$$(P) \quad [\leq](p_1, \dots, p_n) \rightarrow \\ \left(\bigvee_{\langle i_1, \dots, i_j \rangle \in [1, n]} \Diamond(p_{i_1} \wedge \dots \wedge p_{i_j}) \wedge \Box(p_{i_1} \wedge \dots \wedge p_{i_j} \rightarrow p) \right) \leftrightarrow \text{PKB}_l(p)$$

où la modalité \Diamond est le dual de \Box ($\Diamond p \stackrel{\text{def}}{=} \neg \Box \neg p$). La formule $\Diamond p$ doit être lue : p est consistant dans le calcul des propositions.

L'axiome ci-dessus exprime donc qu'un utilisateur de niveau d'habilitation l a la permission de connaître toute formule p dès lors que p est dérivable d'un ensemble consistant de formules p_{i_1}, \dots, p_{i_j} , le niveau de classification des p_{i_1}, \dots, p_{i_j} étant inférieur ou égal à l . En particulier, il est facile de vérifier qu'un utilisateur de niveau d'habilitation l a la permission de connaître toute formule consistante p_i ayant un niveau de classification inférieur ou égal à l .

La sémantique correspondant à ces axiomes a été donnée dans [CUP 96b]. Il est intéressant de remarquer que les notions de permission et d'interdiction ainsi définies, sont indépendantes, dans le sens où ce qui est permis n'est pas nécessairement le complémentaire de ce qui est interdit.

5.3. Application : Expression de propriétés de sécurité

Dans cette section, nous allons définir dans le cadre de la logique Σ présentée dans la section précédente les conditions qui caractérisent les états acceptables du point de vue d'une politique de sécurité multi-niveaux.

Dans la suite, nous utiliserons les notations suivantes :

$$R_l = [\leq](p_1, \dots, p_n) \wedge [>](p'_1, \dots, p'_m)$$

La formule R_l représente la politique de sécurité s'appliquant au niveau l .

$$R = \bigwedge_{l \in \text{Niveau}} R_l$$

La formule R représente la politique de sécurité multi-niveaux globale.

5.3.1. Contrôle des flux d'informations

Les contraintes dites de contrôle des flux d'informations visent à caractériser les situations qui ne violent pas la politique de sécurité. En fait, notre formalisation permet de représenter deux attitudes différentes.

Etant donnée une situation où l'utilisateur u a obtenu de la base de données l'information p , la première attitude considère que cette situation est acceptable si p n'est pas une information interdite. Dans notre logique, cette première attitude s'exprime par la contrainte suivante :

$$(CL) \quad \vdash_{\Sigma} R \rightarrow (KB_u p \rightarrow \neg FKB_I p)$$

où l représente le niveau d'habilitation de u . Cette première attitude est celle utilisée dans toutes les bases de données contenant des informations secrètes et qui essayent de fournir des statistiques sur ces données secrètes à des utilisateurs qui ne sont pas eux-mêmes habilités à connaître ces données secrètes. Dans ce cas, la contrainte ci-dessus indique qu'une situation est acceptable si les données fournies à l'utilisateur ne lui permette pas de déduire les données secrètes. On peut qualifier cette première attitude de libérale.

La deuxième attitude considère acceptables les situations pour lesquelles tout utilisateur ne connaît que des informations permises. Dans notre formalisme, cette seconde attitude correspond à la contrainte suivante :

$$(CC) \quad \vdash_{\Sigma} R \rightarrow (KB_u p \rightarrow PKB_I p)$$

où l représente le niveau d'habilitation de u . Cette seconde attitude est celle utilisée dans tous les SGBD multi-niveaux actuellement disponibles : un utilisateur ne peut accéder à une information de la base que si cette information est effectivement classée à un niveau inférieur au niveau d'habilitation de l'utilisateur. On peut qualifier cette seconde attitude de conservatrice. Il est toutefois intéressant de remarquer que cette seconde contrainte n'empêche pas forcément de connaître des informations interdites. Pour cela, une troisième contrainte doit être satisfaite.

5.3.2. Cohérence de la réglementation

La troisième contrainte concerne la cohérence de la réglementation. Cette contrainte exprime que dans le cadre d'une réglementation, pour aucune formule p il ne peut être à la fois permis et interdit de connaître p , soit :

$$(CR) \quad \vdash_{\Sigma} R \rightarrow \neg(PKB_I p \wedge FKB_I p)$$

Il est intéressant de remarquer que la combinaison des contraintes (CC) et (CR) permet de dériver la contrainte (CL). On assure ainsi qu'aucun utilisateur ne pourra connaître d'information interdite.

5.3.3. Autres contraintes

Dans [CUP 96b], nous avons étudié comment raffiner la contrainte (CC) pour prendre en compte les situations où un utilisateur parviendrait à inférer une information non autorisée en construisant un raisonnement abductif. Informellement, q peut être déduit de p par un raisonnement abductif s'il suffit de rajouter à p l'hypothèse h pour pouvoir déduire q de $\{p, h\}$ par un raisonnement déductif classique. Dans la suite, le fait que q peut être déduit par abduction de p avec l'hypothèse h sera noté $ABD(p, h, q)$ ¹¹.

Si l'on veut considérer les possibilités de raisonnement abductif, la contrainte de sécurité doit exprimer que si q peut être déduit par abduction de p avec l'hypothèse h alors, si l'utilisateur u habilité au niveau l sait que la base de données croit p , alors ou bien tout utilisateur de niveau l a la permission de connaître q , ou bien tout utilisateur de niveau l a l'interdiction de connaître h , soit:

$$(CSA) \quad \vdash_{\Sigma} R \rightarrow (KB_u p \wedge ABD(p, h, q) \rightarrow PKB_l q \vee FKB_l h)$$

Cette contrainte considère comme acceptable une situation où u connaît p , même si p informe sur l'information secrète q , car l'information h que doit acquérir u pour connaître q est secrète.

Dans [CUP 94], nous avons également étudié comment adapter les contraintes de sécurité aux bases de données multi-niveaux qui utilisent des leurres.

Enfin, de nombreuses bases de données multi-niveaux considèrent que les niveaux de classification des informations sont définis par les utilisateurs eux-mêmes au fur et à mesure que les informations sont insérées dans la base de données. Par exemple, un utilisateur pourra choisir un niveau de session (à condition que ce niveau soit inférieur au niveau d'habilitation de l'utilisateur) et toutes les informations introduites par l'utilisateur au cours de cette session seront implicitement classifiées au niveau de la session. Un tel fonctionnement peut être formalisé en ajoutant la contrainte suivante :

$$[l]p \leftrightarrow E_u B_l p$$

où E_u est la modalité de faire en sorte que déjà utilisé dans la section 3.2. Cette contrainte exprime que la classification de p est égal à l si et seulement si il existe un utilisateur u qui a fait en sorte que la base de données de niveau l croit que p est vrai. A partir de ce type de contrainte, un modèle de sécurité, dit modèle de causalité, a été défini [BIE 92].

6. Conclusion

Après avoir présenté sommairement les logiques modales, nous avons montré comment les utiliser comme outil de modélisation pour trois types de problèmes. Nous

11. Remarquons que la définition de $ABD(p, h, q)$ doit être affinée pour rejeter le cas trivial où $\{p, h\}$ serait inconsistent, car dans ce cas, on peut déduire n'importe quelle formule de $\{p, h\}$. Il faut aussi rejeter l'autre cas trivial où h tout seul suffirait à déduire q .

voudrions souligner ici que ces modélisations, qui sont orientées vers les applications, peuvent servir de référence pour la spécification formelle de logiciels qui vont implanter des fonctions telles que : génération de réponses coopératives, fusion de bases, ou protection de la confidentialité. Cet outil de modélisation oblige à résoudre des questions qui n'apparaissent pas spontanément quand on utilise des langages de spécification habituels. Par exemple : que peut-on déduire du fait qu'un usager a la permission de modifier l'information p ou de modifier l'information q? qu'il a la permission de modifier p **ou** la permission de modifier q? ou bien, qu'il a la permission de modifier p **et** la permission de modifier q? ou rien du tout?

D'autre part nous avons à peine évoqué les possibilités de traitements automatiques de ces modèles. Pour la réalisation de maquettes, on peut mentionner la réalisation en PROLOG d'une maquette qui permet de fusionner des bases de données [CHO 93] [CHO 95a], et une autre qui permet de raisonner sur la fiabilité des réponses aux questions en fonction de la fiabilité des sources [CHO 94d]. L'idée générale est de représenter les modalités par des méta-prédicats. Dans un autre ordre d'idées, dans [DEM 96b], nous avons défini une méthode qui, pour une question formulée dans l'algèbre relationnelle, génère deux questions, dans le même langage, telles qu'on peut garantir que la réponse valide se trouve encadrée (pour la relation d'inclusion) par les réponses à ces deux questions. La génération des deux questions définissant cet encadrement est définie par des clauses de Horn qui peuvent s'implanter en PROLOG. Plus ces clauses permettent de définir un encadrement précis, plus le coût de leur génération augmente. Le compromis est à choisir par le concepteur. La traduction des deux questions en SQL ne pose pas de problème, et permet l'utilisation de systèmes existants.

Enfin, nous concluons en disant que les logiques modales, déjà bien connues dans le domaine de l'intelligence artificielle comme puissant outil de formalisation, ont de nombreuses applications possibles dans le domaine des bases de données, qui restent, pour la plus grande part, à explorer.

7. Bibliographie

- [AMO 95] AMO S. D., « *Contraintes dynamiques et schémas transactionnels* ». PhD thesis, Université Paris XIII, 1995.
- [BID 93] BIDOIT N. et AMO S. D., « Contraintes dynamiques d'inclusion et schémas transactionnels ». In *Neuvièmes journées bases de données avancées*, 1993.
- [BID 95] BIDOIT N. et AMO S. D., « A first step towards implementing dynamic algebraic dependencies ». In *Proc of ICDT*, 1995.
- [BIE 92] BIEBER P. et CUPPENS F., « A Logical View of Secure Dependencies ». *Journal of Computer Security*, vol. 1, n° 1, p. 99–129, 1992.
- [BON 92] BONATTI P. A., KRAUS S. et SUBRAHMANIAN V., « Declarative foundations of secure deductive databases ». In *Proc of ICDT*, 1992.
- [BON 95] BONATTI P. A., KRAUS S. et SUBRAHMANIAN V., « Foundations of secure deductive databases ». *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 7, n° 3, 1995.
- [CAS 82a] CASANOVA M. et FURTADO A., « A family of temporal languages for the description of transition constraints ». In H. GALLAIRE J. M. et NICOLAS J., Eds., *Advances in database theory*, 1982.
- [CAS 82b] DE CASTILHO J., CASANOVA M. et FURTADO A., « A temporal framework for database specifications ». In *proceedings of VLDB*, 1982.
- [CAS 84] CASANOVA M. et FURTADO A. L., « *On the description of database transition constraints using temporal constraints* ». Plenum Press, 1984.

- [CAZ 92] CAZALENS S. et DEMOLOMBE R., « Intelligent access to data and knowledge bases via users' topics of interest ». In *Proceedings of IFIP Conference*, p. 245–251, 1992.
- [CHE 88] CHELLAS B. F., *Modal Logic: An introduction*. Cambridge University Press, 1988.
- [CHE 95] CHELLAS B., « On bringing it about ». *Journal of Philosophical Logic*, vol. 24, 1995.
- [CHO 86a] CHOLVY L., « A modal approach to update semantics problem ». In *Proceedings of the second IFIP 2.6 Working Conference on Database Semantics (DS-2)*. North Holland, 1986.
- [CHO 86b] CHOLVY L., « Update semantics under the domain closure assumption ». In *Proceedings of International Conference on Database Theory (ICDT)*, Rome, 1986.
- [CHO 92] CHOLVY L., « A logical approach to multi-sources reasoning ». In *Proceedings of the Applied Logic Conference*, Amsterdam, december 1992.
- [CHO 93] CHOLVY L., « Proving theorems in a multi-sources environment ». In *Proceedings of IJCAI*, p. 66–71, 1993.
- [CHO 94a] CHOLVY L., « Fusion de sources d'informations contradictoires ordonnées en fonction des thèmes ». *Revue de l'Intelligence artificielle*, vol. 8, n° 2, 1994.
- [CHO 94b] CHOLVY L., A logical approach to multi-sources reasoning. In *Lecture notes in Artificial Intelligence*, n° 808. Springer-Verlag, 1994.
- [CHO 94c] CHOLVY L. et DEMOLOMBE R., « Reasoning with information sources ordered by topics ». In *Proceedings of of Artificial Intelligence: Methods, Systems and Applications (AIMSA)*. World Scientific, Sofia, september 1994.
- [CHO 94d] CHOLVY L., DEMOLOMBE R. et JONES A., Reasoning about the safety of information : from logical formalization to operational definition. In *Lecture notes in Artificial Intelligence*, n° 869. Springer-Verlag, 1994.
- [CHO 95a] CHOLVY L., « Automated reasoning with merged contradictory information whose reliability depends on topics ». In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU)*, Fribourg, July 1995.
- [CHO 95b] CHOLVY L. et CUPPENS F., Providing Consistent Views in a Polyinstantiated Database. In BISKUP J., MORGENSTERN M. et LANDWEHR C., Eds., *Database Security, 8: Status and Prospects*. North-Holland, 1995. Results of the IFIP WG 11.3 Workshop on Database Security.
- [CHO 96] CHOLVY L., « Answering queries addressed to a group of deductive databases ». In *Workshop on flexible query-answering*. Roskilde University, 1996.
- [CUP 88] CUPPENS F. et DEMOLOMBE R., « Cooperative Answering: a methodology to provide intelligent access to Databases ». In *Proc of Expert Database Systems*, 1988.
- [CUP 94] CUPPENS F. et DEMOLOMBE R., « Normative Conflicts in a Confidentiality Policy ». In *ECAI'94 Workshop on Artificial Normative Reasoning*, Amsterdam, The Netherlands, 1994.
- [CUP 96a] CUPPENS F., « Querying a Multilevel Database: A Logical Analysis ». In *Proceedings of the 22nd International Conference on Very Large Data Bases*, Bombay, India, 1996.
- [CUP 96b] CUPPENS F. et DEMOLOMBE R., « A Deontic Logic for Reasoning about Confidentiality ». In *Proc. of 3rd International Workshop on Deontic Logic in Computer Science*, 1996.
- [DEM 94a] DEMOLOMBE R. et IMIELINSKI T., *Nonstandard Queries and Nonstandard Answers*. Oxford University Press, 1994.
- [DEM 94b] DEMOLOMBE R. et JONES A., « Deriving answers to safety queries ». In DEMOLOMBE R. et IMIELINSKI T., Eds., *Non standard queries and answers*. Oxford University Press, 1994.
- [DEM 95] DEMOLOMBE R. et JONES A., « Reasoning about Topics: towards a formal theory ». In *American Association for Artificial Intelligence Fall Symposium*, 1995.
- [DEM 96a] DEMOLOMBE R., « Answering queries about validity and completeness of data: from Modal Logic to Relational Algebra ». In *Proc. of the Workshop on Flexible Query-Answering Systems*. Roskilde University, 1996.
- [DEM 96b] DEMOLOMBE R., « Validity Queries and Completeness Queries ». In *Proc. of 9th International Symposium on Methodologies for Intelligent Systems*, 1996.
- [DEM 96c] DEMOLOMBE R., JONES A. J. I. et CARMO J., « Toward a uniform logical representation of different kinds of Integrity Constraints ». In CONRAD S., KLEIN H.-J. et SCHEWE K.-D., Eds., *Integrity in Databases*. Otto von Guericke Universitat Magdeburg, 1996.

- [DEM 96d] DEMOLOMBE R. et JONES A., « Integrity Constraints Revisited ». *Journal of the Interest Group in Pure and Applied Logics*, vol. 4, n° 3, 1996.
- [DEM 96e] DEMOLOMBE R. et JONES A., « On sentences of the kind “sentence “p” is about topic “t” : some steps toward a formal-logical analysis ». In H-J. OHLBACH AND U. REYLE, Ed., *Logic, Language and Reasoning. Essays in Honor of Dov Gabbay*. Kluwer Academic Press, 1996.
- [DEN 82] DENNING D., *Cryptography and Data Security*. Addison-Wesley, 1982.
- [FAR 85] FARINAS L., « MÓLOG : A system that extends PROLOG with modal logic ». Rapport technique, Université Paul Sabatier, Toulouse, France, 1985.
- [GAL 78] GALLAIRE H. et MINKER J., *Logic and databases*. Plenum, 1978.
- [GAL 81] GALLAIRE H., MINKER J. et NICOLAS J.-M., *Advances in Data Base Theory, Vol1*. Plenum Press, 1981.
- [GAL 83] GALLAIRE H., MINKER J. et NICOLAS J.-M., *Advances in Data Base Theory, Vol2*. Plenum Press, 1983.
- [GAL 84] GALLAIRE H., MINKER J. et NICOLAS J. M., « Logic and databases : a deductive approach ». *ACM Surveys*, vol. 16, n° 2, 1984.
- [HUG 72] HUGHES G. E. et CRESSWELL M. J., « *An introduction to modal logic* ». Methuen London and New York, 1972.
- [KUN 84] KUNG C., « A temporal framework for database specification and verification ». In *proceedings of VLDB*, 1984.
- [KUN 85] KUNG C., « On verification of database temporal constraints ». In *Proc of PODS*, 1985.
- [POR 77] PORN I., « Action Theory and Social Science. Some Formal Models ». *Synthese Library*, vol. 120, 1977.
- [REI 88] REITER R., « What should a database know ». In *Proc of PODS*, 1988.
- [REI 92] REITER R., « On formalizing database updates ». In *Proc of EDBT*, 1992.