Bases logiques

Master 2, Info, F3 (IA et Image), RATS: partie approches logiques (2005)

1 Introduction

(à ces “bases logiques”, et aussi à cette partie “approches logiques” du cours RATS)

La logique formelle possède l’intérêt d’être un langage universel, qui constitue un bon compromis entre une langue naturelle et un langage directement adapté aux ordinateurs. Il est de toute façon indispensable de connaître les rudiments, et en particulier d’avoir une idée de ce que représente la distinction entre les aspects *syntaxiques* et *sémantiques*. En effet, ces notions apparaissent dans de nombreux domaines de l’informatique. En ce qui concerne les aspects représentation des connaissances et intelligence artificielle, l’apport de la logique est encore plus important. Le langage même de la logique formelle classique est souvent utilisé, plus ou moins directement, pour représenter et pour traiter l’information. Cela permet d’éviter à l’utilisateur d’apprendre un langage trop technique, et trop particulier. L’universalité du langage de la logique formelle a en effet deux facettes utiles ici. D’une part, cela signifie que l’effort nécessaire pour utiliser une variante précise de la logique classique, adaptée à un type de problème donné, ne sera pas trop important. D’autre part, une méthode qui a permis de résoudre un problème précis pourra s’adapter assez directement à d’autres domaines.

Toutefois, on s’est vite aperçu qu’il convient d’étendre ou de particulariser la logique classique, dès que l’on veut traiter un problème réel non “jouet”. L’ambition de départ était la suivante: on peut tout exprimer (ce qui est calculable par un ordinateur) en logique classique, et comme on saura vite trouver des démonstrateurs universels efficaces, cela suffira. Les problèmes de complexité rendent en fait cette méthode inapplicable: les calculs nécessitent très vite un temps et un espace de stockage rédhibitoires. Cela rend nécessaire des langages intermédiaires, adaptés à des classes de problèmes précis. Ces langages ont un double but: faciliter la traduction directe d’un problème précis par l’utilisateur, et faciliter le calcul effectif par l’ordinateur. Ces langages sont donc toujours des compromis entre deux exigences contradictoires: (1) une grande expressivité et, ce qui est aussi important, une grande facilité à traduire un problème précis dans ce langage, (2) un aspect suffisamment contraignant pour forcer l’utilisateur à traduire son problème d’une façon qui va permettre le calcul automatique efficace. C’est en grande partie à cause de ces exigences contradictoires qu’il n’existe pas à ce jour de solution universelle, et qu’un grand nombre de propositions coexistent. Ce domaine est très actif et en évolution très rapide, mais personne ne semble plus penser aujourd’hui qu’une solution universelle soit pour bientôt. Il faut donc faire avec les propositions existantes.

Ce cours a pour but de familiariser avec ce sujet. Il contient (sauf (1d), donné uniquement pour la “culture”):

1. Un rappel (?) des bases de la logique formelle (support papier uniquement, à étudier tranquillement, pratiquement pas de transparents pour ce point).
   (a) *Syntaxe* et *sémantique* (définitions et résultats de base), pour la logique classique: *propositionnelle*, et
   (b) *calcul des prédicats* (*premier ordre* uniquement ici).
   (c) Une sensibilisation aux extensions les plus fréquentes à la logique classique, liées au problème de la *minimisation des informations positives* (“on ne peut pas demander à l’utilisateur d’exprimer tout ce qui n’est pas vrai”): *hypothèse du monde clos*, *complétion de prédicats*, *circonscription*. Ces extensions à la logique classique n’existaient pas avant l’apparition des ordinateurs, mais elles se sont avérées nécessaires et sont maintenant des “classiques” dont il faut au moins avoir une idée. La circonscription intervient explicitement dans la présentation faite dans ce cours du calcul des événements (version de Shanahan).
   (d) Une sensibilisation aux *logiques modales*. Ces logiques existaient bien avant l’informatique (on peut les faire remonter jusqu’à Aristote), mais elles ont connu un développement considérable depuis l’apparition des ordinateurs. Le vocabulaire (*sémantique des mondes possibles* à a Kripke, etc...) et les principes de base doivent être connus si on veut comprendre la littérature actuelle en représentation de connaissances, principalement pour ce qui concerne les actions et le temps. En particulier, des logiques modales qui traitent le temps ou le spatial “qualitatif” sont très utilisées actuellement.

2. Une présentation de deux types de formalismes actuellement utilisés pour représenter les situations évolutives:

   (a) Le calcul situationnel, issu des premiers travaux sur la minimisation des informations positives. On présente essentiellement ici la version de Reiter et al.

   (b) Le calcul des événements, issu de la programmation logique, après avoir tendu à s'en détacher, tend aujourd'hui à s'en rapprocher à nouveau grâce aux approches type "answer sets" de la programmation logique. On présente essentiellement ici la version de Shanahan, fondée sur la circonscription.

2 Logique propositionnelle

2.1 Préliminaires

Un langage de logique propositionnelle $\mathbb{L}$ a pour vocabulaire:

- les *connecteurs* $\neg$, $\wedge$, $\vee$, $\Rightarrow$ (on peut trouver aussi $\rightarrow$ ou $\supset$ pour l'implication), $\Leftrightarrow$, (on trouve aussi, plus rarement, $\oplus$ pour le "ou exclusif": $\varphi \oplus \psi$ équivaut à $\neg(\varphi \Leftrightarrow \psi)$, soit $\varphi \not\Leftrightarrow \psi$,

- un certain nombre de *symboles propositionnels* $A, B, C, \ldots, A_i, \ldots$ souvent, on y ajoute deux symboles particuliers $VRAI$ et $FAUX$.

- on utilise aussi des parenthèses dans l'écriture des formules.

L'ensemble des *formules propositionnelles* de $\mathbb{L}$ est défini ainsi:

- tout symbole propositionnel de $\mathbb{L}$ est une formule,

- si φ est une formule, $\neg\varphi$ aussi,

- si φ et ψ sont des formules, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \Rightarrow \psi$, $\varphi \Leftrightarrow \psi$ aussi.

2.2 Théorie des modèles, ou *sémantique* (donner du *sens* aux formules)

Une *interprétation* μ d'un langage propositionnel $\mathbb{L}$ est une application de l'ensemble de tous les symboles propositionnels de $\mathbb{L}$ dans $\{VRAI, FAUX\}$[1].

Souvent, on note aussi une interprétation comme le sous-ensemble de l'ensemble des symboles propositionnels de $\mathbb{L}$ composé des symboles A tels que $\mu(A) = VRAI$.

La *relation de satisfaction* $\models$ est l'application de l'ensemble des interprétations dans l'ensemble de toutes les formules de $\mathbb{L}$, définie inductivement ainsi (où A désigne un symbole propositionnel, μ une interprétation et φ et ψ des formules quelconques de $\mathbb{L}$):

- $\mu \models A$ si $\mu(A) = VRAI$.

- $\mu \models \neg\varphi$ si $\mu \not\models \varphi$.

- $\mu \models \varphi \wedge \psi$ si $\mu \models \varphi$ et $\mu \models \psi$,

- $\mu \models \varphi \vee \psi$ si $\mu \models \varphi$ ou $\mu \models \psi$,

- $\mu \models \varphi \Rightarrow \psi$ si $\mu \models \neg\varphi \vee \psi$,

- $\mu \models \varphi \Leftrightarrow \psi$ si $\mu \models \varphi \Rightarrow \psi$ et $\mu \models \psi \Rightarrow \varphi$.

[1] Par un abus commode, on note de la mˆeme faç on les deux formules sp´eciales et ces valeurs de v´erit´e.

Une interprétation μ *satisfait* une formule φ, ou est un *modèle* de φ, si $\mu \models \varphi$.
μ *falsifie* φ si $\mu \not\models \varphi$.

Exemple: L'ensemble des symboles propositionnels de $\mathbb{L}$ est $\{A,B,C,D\}$, et μ est définie par

	A	B	C	D
μ	$VRAI$	$FAUX$	$VRAI$	$FAUX$

On note $\mu = \{A,C\}$, ou plus rarement $\mu = \{A,\not B,C,\not D\}$.

$\mu \models ((A \Rightarrow C) \wedge (B \Rightarrow (A \wedge C)))$, $\qquad \mu \not\models ((A \wedge C) \Rightarrow (B \vee D))$.

Remarque: Pour toute formule φ et toute interprétation μ, on a soit $\mu \models \varphi$ soit $\mu \models \neg\varphi$.

Une *théorie* $\mathcal{T}$ de $\mathbb{L}$ est un ensemble de formules de $\mathbb{L}$. Une interprétation μ *satisfait* une théorie $\mathcal{T}$, ou est un *modèle* de $\mathcal{T}$ si $\mu \models \varphi$ pour toute formule φ de $\mathcal{T}$.

Une formule est *valide* si elle est satisfaite par toute interprétation.

Une formule ou une théorie est *insatisfiable* si elle n'est satisfaite par aucune interprétation.

Par définition, la formule $VRAI$ (aussi notée $\top$) est valide, la formule $FAUX$ (ou $\bot$) est insatisfiable.

On définit la relation de conséquence sémantique, notée également $\models$, de l'ensemble des théories de $\mathbb{L}$ dans l'ensemble des formules de $\mathbb{L}$ par: $\quad \mathcal{T} \models \varphi$ si tout modèle de $\mathcal{T}$ satisfait φ.

On note $Th(\mathcal{T})$ l'ensemble de toutes les conséquences (sémantiques) de $\mathcal{T}$: $\varphi \in Th(\mathcal{T})$ ssi $\mathcal{T} \models \varphi$.

Théorème de la déduction:

$$\mathcal{T}, \psi \models \varphi \quad \text{ssi} \quad \mathcal{T} \models \psi \Rightarrow \varphi.$$

($\mathcal{T}, \psi$ est une abréviation commode de $\mathcal{T} \cup \{\psi\}$, de même on abrège souvent $\mathcal{T} \cup \mathcal{T}'$ en $\mathcal{T}, \mathcal{T}'$.)

Théorème de compacité

$$\mathcal{T} \models \varphi \text{ ssi il existe un sous-ensemble fini } \mathcal{T}' \subseteq \mathcal{T} \text{ tel que } \mathcal{T}' \models \varphi.$$

Attention: Les notations données ci-dessus sont standards, mais ambiguës. En général, le contexte indique le sens à donner aux symboles. Toutefois, il faut éviter de confondre le "$\models$" de la relation de satisfaction et le "$\models$" de la conséquence sémantique. Si $\{A,C\}$ désigne un modèle (employé ici au sens d'interprétation: c'est toujours le modèle de quelque théorie), on évitera d'écrire directement $\{A,C\} \models \ldots$. On écrira plutôt: "soit l'interprétation $\mu = \{A,C\}$, on a $\mu \models \neg B$" (par exemple, si le langage $\mathbb{L}$ comprend aussi le symbole B). La notation $\mu = \{A,\not B,C\}$ prête un peu moins à confusion, mais n'est utilisable que si $\mathbb{L}$ contient peu de symboles propositionnels (c'est pourquoi elle n'est que très rarement employée).

L'écriture $\{A,C\} \models \ldots$ devra être réservée au cas où $\{A,C\}$ désigne une théorie. L'ambiguïté provient aussi du fait que l'on note de la même façon le symbole propositionnel A et la formule A. Si $\mathcal{T} = \{A,C\}$ est une théorie, on n'a pas $\mathcal{T} \models \neg B$: $\mathcal{T}$ possède plusieurs modèles, et tous ne satisfont pas $\neg B$. De même (voir partie 2.3 ci-dessous), il est impossible de trouver une démonstration de la formule $\neg B$ à partir des deux seules formules A et C.

Petite question: si le langage $\mathbb{L}$ comporte seulement les trois symboles propositionnels A, B et C, quels sont tous les modèles de $\mathcal{T}$? Que devient le nombre de modèles de $\mathcal{T}$ chaque fois qu'on ajoute un symbole propositionnel à $\mathbb{L}$?

Remarque: il est très fréquent de n'avoir ni $\mathcal{T} \models \varphi$ ni $\mathcal{T} \models \neg\varphi$.

Une théorie est *complète* dans $\mathbb{L}$ si, pour toute formule φ de $\mathbb{L}$, on a $\mathcal{T} \models \varphi$ ou $\mathcal{T} \models \neg\varphi$.

On utilise aussi la notion suivante: Si μ est une interprétation de $\mathbb{L}$, la *théorie de* μ, notée également $Th(\mu)$, est l'ensemble de toutes les formules de $\mathbb{L}$ qui sont satisfaites par μ. Il s'agit toujours d'une théorie complète.

Exemple: soit l'interprétation $\mu = \{A,\not B,C\}$, la théorie de μ est $Th(A,\neg B,C) = Th(A \wedge \neg B \wedge C)$.

2.3 Théorie de la démonstration (aspect *syntaxique*: manipuler des formules)

Un exemple de système axiomatique pour la logique propositionnelle

- Schémas d'axiomes

  1. $(\varphi \vee \varphi) \Rightarrow \varphi$
  2. $\varphi \Rightarrow (\varphi \vee \psi)$
  3. $(\psi \vee \varphi) \Rightarrow (\varphi \vee \psi)$
  4. $(\psi \Rightarrow \varphi) \Rightarrow ((\chi \vee \psi) \Rightarrow (\chi \vee \varphi))$

- Règle d'inférence $\dfrac{\varphi \qquad \varphi \Rightarrow \psi}{\psi}$ (Modus Ponens)

- Définition de connecteurs: $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\quad \varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi$, $\quad \varphi \Leftrightarrow \psi \equiv (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$.

Afin d'améliorer la lisibilité, on a utilisé le connecteur $\Rightarrow$ dans l'énoncé des axiomes, alors qu'il n'est introduit qu'après. Rigoureusement, il aurait fallu écrire $\neg(\varphi \vee \varphi) \vee \varphi$, etc... $\equiv$ est ici un *metasymbole* (pas dans le langage logique) signifiant "équivalent à", on aurait pu ici le remplacer par $=_{d\acute{e}f}$("égal par définition à").

Dans ce système axiomatique, une formule φ *se déduit* d'une théorie $\mathcal{T}$, ce qu'on note $\mathcal{T} \vdash \varphi$, s'il existe une suite finie de formules $\psi_1, \cdots, \psi_n$ telle que ψ_n est φ et, pour tout $i \in \{1, \cdots, n\}$:

- soit $\psi_i \in \mathcal{T}$,
- soit ψ_i est une instance d'un des schémas d'axiomes donnés ci-dessus,
- soit ψ_i résulte de l'application du modus ponens à deux formules ψ_j, ψ_k où $j, k < i$.

Si une telle suite existe, on dit qu'on a trouvé une *déduction de* φ *à partir de* $\mathcal{T}$.

Théorème de validité (ou correction, ou adéquation) **et de complétude:**
Le système décrit ci-dessus est *valide* et *complet* pour la logique propositionnelle:
on a (où le sens "si" $\hookleftarrow$ correspond à la validité et le sens "seulemnt si" $\hookrightarrow$ à la complétude)

$$\mathcal{T} \vdash \varphi \qquad \text{ssi} \qquad \mathcal{T} \models \varphi.$$

Remarques: Pour démontrer rigoureusement $\mathcal{T} \vdash \varphi$ (c'à-d $\varphi \in Th(\mathcal{T})$), le plus simple est en général de trouver une déduction de φ à partir de $\mathcal{T}$.

Une autre solution utilisée parfois en démonstration automatique pour démontrer $\mathcal{T} \vdash \varphi$ consiste à démontrer $\mathcal{T}, \neg\varphi \vdash FAUX$ (cf théorème de la déduction, sachant que φ équivaut à $\neg\varphi \Rightarrow FAUX$), ce qui revient à démontrer que $\mathcal{T}, \neg\varphi$ n'a pas de modèle (cf théorème de validité et de complétude).

Pour démontrer $\mathcal{T} \not\vdash \varphi$, le plus simple est en général de trouver un modèle de $\mathcal{T}, \neg\varphi$.
En particulier $\mathcal{T}$ est *consistante* ou *cohérente* ($\mathcal{T} \not\vdash FAUX$) ssi elle est satisfiable ($\mathcal{T}$ a un modèle).

Le système présenté ci-dessus est élégant, mais peu adapté à la démonstration automatique car même les démonstrations les plus simples peuvent être très longues. En général, les démonstrateurs utilisent d'autres axiomes et d'autres règles.

Une règle fréquemment utilisée est la *règle de résolution:* $\dfrac{\varphi \vee \psi \qquad \varphi \vee \neg\psi}{\varphi}$.

Un cas particulier important est $\dfrac{\psi \qquad \neg\psi}{FAUX}$.

On peut même ne considérer que des clauses comme formules φ, ψ car toute formule équivaut à un ensemble fini de clauses. (Un *atome* A est une formule réduite à un symbole propositionnel, un *littéral* est un atome A ou sa négation $\neg A$, une *clause* est une disjonction de littéraux $A \vee B \vee \neg C \ \vee \cdots$). $FAUX$ est alors appelée la *clause vide* (disjonction sur 0 terme).

2.4 Quelques formules valides (ou *tautologies*) utiles

- Associativité et commutativité de $\wedge$ et de $\vee$.
- Distributivités:
$$(\varphi \wedge (\psi \vee \chi)) \Leftrightarrow ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$$
$$(\varphi \vee (\psi \wedge \chi)) \Leftrightarrow ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$$
- Négation:
$$(\neg\neg\varphi) \Leftrightarrow \varphi$$
$$(\neg(\varphi \Rightarrow \psi)) \Leftrightarrow (\varphi \wedge \neg\psi)$$
$$(\neg(\varphi \Leftrightarrow \psi)) \Leftrightarrow ((\varphi \wedge \neg\psi) \vee (\neg\varphi \wedge \psi)) \qquad \textit{(ou exclusif)}$$
- Lois de De Morgan:
$$(\neg(\varphi \wedge \psi)) \Leftrightarrow (\neg\varphi \vee \neg\psi)$$
$$(\neg(\varphi \vee \psi)) \Leftrightarrow (\neg\varphi \wedge \neg\psi)$$
- Divers:

tiers exclus, contradiction	$\varphi \vee \neg\varphi$	et	$\neg(\varphi \wedge \neg\varphi)$
implication	$(\varphi \Rightarrow (\psi \Rightarrow \varphi))$	et aussi	$(\varphi \Rightarrow (\neg\varphi \Rightarrow \psi))$
contraposition	$(\varphi \Rightarrow \psi)$	$\Leftrightarrow$	$(\neg\psi \Rightarrow \neg\varphi)$
chaîne d'implications	$(\varphi \Rightarrow (\psi \Rightarrow \chi))$	$\Leftrightarrow$	$((\varphi \wedge \psi) \Rightarrow \chi)$
conjonctions d'implications	$((\varphi \Rightarrow \chi) \wedge (\psi \Rightarrow \chi))$	$\Leftrightarrow$	$((\varphi \vee \psi) \Rightarrow \chi)$
(très utiles, cf Prolog,...)	$((\varphi \Rightarrow \psi) \wedge (\varphi \Rightarrow \chi))$	$\Leftrightarrow$	$(\varphi \Rightarrow (\psi \wedge \chi))$
disjonctions d'implications	$((\varphi \Rightarrow \chi) \vee (\psi \Rightarrow \chi))$	$\Leftrightarrow$	$((\varphi \wedge \psi) \Rightarrow \chi)$
	$((\varphi \Rightarrow \psi) \vee (\varphi \Rightarrow \chi))$	$\Leftrightarrow$	$(\varphi \Rightarrow (\psi \vee \chi))$.

3 Logique des prédicats du premier ordre

3.1 Préliminaires

Un langage de logique du premier ordre $\mathcal{L}$ a pour vocabulaire, en plus des connecteurs $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$:

- les *quantificateurs* $\exists$, $\forall$,
- des *variables (d'individus)* $x, y, z, \ldots$ (en nombre suffisant ...),
- un certain nombre de symboles de *fonctions d'arité* $i = 0, 1, 2, \cdots$: $\quad f, g, \cdots$,
  les fonctions d'arité nulle sont des *constantes*,
- un certain nombre de symboles de *prédicats d'arité* $i = 0, 1, 2, \cdots$: $\quad P, Q, \cdots$,
  les prédicats d'arité nulle sont des symboles propositionnels,
  l'égalité $(=)$ est un symbole de prédicat d'arité 2 particulier,
- on utilise aussi des parenthèses dans l'écriture des formules.

L'ensemble des *termes* de $\mathcal{L}$ est défini ainsi:

- tout symbole de variable est un terme,
- si f est un symbole de fonction d'arité i de $\mathcal{L}$ et si $t_1, \cdots, t_i$ sont i termes de $\mathcal{L}$, $f(t_1, \cdots, t_i)$ est un terme; en particulier tout symbole de constante est un terme.

L'ensemble des *formules* de $\mathcal{L}$ est défini ainsi:

- si P est un symbole de prédicat d'arité i de $\mathbb{L}$ et si $t_1, \cdots, t_i$ sont i termes de $\mathbb{L}$, $P(t_1, \cdots, t_n)$ est une formule (*atomique*), en particulier
- toute proposition de $\mathbb{L}$ est une formule atomique, et aussi
- si $\mathbb{L}$ contient le symbole d'égalité, $t_1 = t_2$ est une formule atomique;
- si φ et ψ sont des formules, $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \Rightarrow \psi$ et $\varphi \Leftrightarrow \psi$ sont aussi des formules,
- si φ est une formule, alors $\exists x \varphi$ et $\forall x \varphi$ sont aussi des formules.

Une formule atomique ou sa négation est un *littéral* (respectivement *positif* ou *négatif*).

On note $Var(t)$ l'ensemble des symboles de variables qui apparaissent dans un terme t. Si $Var(t) = \emptyset$, t est un *terme concret*. L'ensemble $Var(\varphi)$ des *variables libres* d'une formule φ est défini ainsi:

- les variables libres d'une formule atomique sont toutes les variables qui y figurent: $Var(P(t_1, t_2, \cdots t_i))$ est $Var(t_1) \cup Var(t_2) \cup \cdots Var(t_i)$ et donc $Var(t_1 = t_2)$ est $Var(t_1) \cup Var(t_2)$.
- $Var(\exists x \varphi)$ est $Var(\varphi) - \{x\}$, de même que $Var(\forall x \varphi)$,
- $Var(\neg\varphi)$ est $Var(\varphi)$, $Var(\varphi * \psi)$ est $Var(\varphi) \cup Var(\psi)$ où $*$ est un des connecteurs binaires $\wedge, \vee, \Rightarrow, \Leftrightarrow$,

Si une occurrence de x est libre dans φ, alors cette occurrence est *liée par le quantificateur* de tête $\forall$ (resp. $\exists$) dans la formule $\forall x \varphi$ (resp. $\exists x \varphi$).

Exemple: la première occurrence de x est libre et la seconde est liée dans la formule φ suivante: $P(x) \wedge \forall x \exists z Q(x,z)$. $Var(\varphi)$ est $\{x\}$ (à éviter, préférer l'écriture équivalente $P(x) \wedge \forall y \exists z Q(y,z)$, on peut toujours renommer une variable liée).

Une formule sans variable libre ($Var(\varphi)$ est vide) est une *formule close* (en anglais: *"sentence"*).

3.2 Théorie des modèles

Une *interprétation* μ d'un langage $\mathbb{L}$ du premier ordre consiste en un ensemble non vide D_μ, le *domaine* de μ, et une fonction d'interprétation $|.|_\mu$. Intuitivement, le domaine est l'ensemble des "objets" de l'interprétation concernée, et la fonction d'interprétation assigne un objet à chaque terme du langage et une valeur de vérité (dans $\{VRAI, FAUX\}$) à chaque formule du langage. L'objet (pour un terme non concret) ou la valeur de vérité (pour une formule non close) peut bien sûr "varier en fonction des variables libres concernées". C'est pourquoi il faut aussi introduire une *fonction d'assignation* dont le rôle est "d'interpréter les variables". Formellement, D_μ et $|.|_\mu$ sont définis ainsi:

- si f est un symbole de fonction d'arité i de $\mathbb{L}$, $|f|_\mu$ est une fonction de D_μ^i dans D_μ, en particulier
- si c est un symbole de constante de $\mathbb{L}$, $|c|_\mu$ est un élément de D_μ,
- si P est un symbole de prédicat d'arité i de $\mathbb{L}$, $|P|_\mu$ est un sous-ensemble de D_μ^i, en particulier
- si P est un symbole propositionnel de $\mathbb{L}$, $|P|_\mu$ est soit $\emptyset$, correspondant à la valeur $FAUX$, soit le singleton D_μ^0, correspondant à la valeur $VRAI$,
- si $\mathbb{L}$ contient le symbole d'égalité, on impose à $|=|_\mu$ d'être la diagonale de $D_\mu \times D_\mu$: $|=|_\mu$ est $\{(e,e) \ / \ e \in D_\mu\}$.

Une *fonction d'assignation* (on dit aussi une *valuation*) des variables dans une interprétation μ de $\mathbb{L}$ est une application v de l'ensemble des variables de $\mathbb{L}$ dans le domaine D_μ de μ.

Pour tout terme t de $\mathbb{L}$, l'interprétation de t dans (μ, v), notée $|t|_\mu^v$, est définie ainsi:

- pour toute variable x, $|x|^\nu_\mu = \nu(x)$,
- si f est un symbole de fonction de $\mathbb{L}$ d'arité i, et si $t_1,\cdots,t_i$ sont i termes de $\mathbb{L}$, $|f(t_1,\cdots,t_i)|^\nu_\mu = |f|_\mu(|t_1|^\nu_\mu,\cdots,|t_n|^\nu_\mu)$, en particulier
- si c est un symbole de constante de $\mathbb{L}$, $|c|^\nu_\mu = |c|_\mu$.

Pour toute formule φ de $\mathbb{L}$, on définit une *relation de satisfaction*, notée $\models$, de la classe des couples (μ,ν) dans l'ensemble des formules de $\mathbb{L}$ ainsi (où les t_i sont des termes, P est un symbole de prédicat d'arité i, et ψ et χ sont des formules, de $\mathbb{L}$):

- $(\mu,\nu) \models P(t_1,\cdots,t_i)$ si $(|t_1|^\nu_\mu,\cdots,|t_i|^\nu_\mu) \in |P|_\mu$, ainsi $(\mu,\nu) \models t_1{=}t_2$ si $|t_1|^\nu_\mu = |t_2|^\nu_\mu$,
- $(\mu,\nu) \models \neg\psi$ si $(\mu,\nu) \not\models \psi$,
- $(\mu,\nu) \models (\psi \wedge \chi)$ si $(\mu,\nu) \models \psi$ et $(\mu,\nu) \models \chi$,
- $(\mu,\nu) \models (\psi \vee \chi)$ si $(\mu,\nu) \models \psi$ ou $(\mu,\nu) \models \chi$,
- $(\mu,\nu) \models (\psi \Rightarrow \chi)$ si $(\mu,\nu) \models (\neg\psi \vee \chi)$, $(\mu,\nu) \models (\psi \Leftrightarrow \chi)$ si $(\mu,\nu) \models (\psi \Rightarrow \chi) \wedge (\chi \Rightarrow \psi)$,
- $(\mu,\nu) \models (\exists x \psi)$ s'il existe un élément e de D_μ et une fonction d'assignation de variables ν', égale à ν sauf que $\nu'(x)$ est égal à e, telle que $(\mu,\nu') \models \psi$,
- $(\mu,\nu) \models (\forall x \psi)$ si $(\mu,\nu) \models \neg(\exists x \neg\psi)$, c'est-à-dire si pour toute fonction d'assignation de variables ν', égale à ν sauf que $\nu'(x)$ peut être n'importe quel élément choisi dans D_μ, on a $(\mu,\nu') \models \psi$.

Si φ est une formule close de $\mathbb{L}$, alors quelles que soient les fonctions d'assignation des variables ν et ν', on a $(\mu,\nu) \models \varphi$ si $(\mu,\nu') \models \varphi$. Si on a $(\mu,\nu) \models \varphi$, on écrit $\mu \models \varphi$: l'interprétation μ satisfait la formule close φ, ou est un *modèle de* φ. Sinon, on écrit $\mu \not\models \varphi$: l'interprétation μ falsifie la formule close φ.

Pour chaque formule close φ et chaque interprétation μ, on a soit $\mu \models \varphi$ soit $\mu \models \neg\varphi$, comme en propositionnel.

Les notions de formules *valides*, et de *conséquence sémantique*, sont définies à partir des relations $\models$ définies ci-dessus exactement comme en propositionnel. En fait, on n'utilisera ici la conséquence sémantique que dans le cas où **toutes les formules (celles de $\mathcal{T}$ et φ) sont closes**: au besoin, en prenant la clôture universelle des formules. La *clôture universelle* de ψ est $\forall x \forall y, \ldots \psi$ où $\{x,y,\ldots\}$ est $Var(\psi)$.

On définit alors la notion de conséquence sémantique en logique du premier ordre, notée elle ausi $\models$, de la façon suivante:

$\mathcal{T} \models \varphi$ si pour toute interprétation μ du langage $\mathbb{L}$ de $\mathcal{T}$, on a: si $\mu \models \mathcal{T}$ alors $\mu \models \varphi$.

Le **théorème de la déduction** demeure:

Si $\mathcal{T} \cup \{\varphi,\psi\}$ est un ensemble de formules closes, on a

$$\mathcal{T},\varphi \models \psi \quad \text{ssi} \quad \mathcal{T} \models \varphi \Rightarrow \psi.$$

Le **théorème de compacité** demeure aussi.

3.3 Système axiomatique de la logique du premier ordre

On peut définir un système axiomatique de la même façon que pour la logique propositionnelle. Il suffit d'ajouter quelques schémas d'axiomes et règles d'inférence. On définit ainsi une relation de conséquence syntaxique $\vdash$. Si $\mathbb{L}$ contient le symbole d'égalité, il faut ajouter des axiomes particuliers, dits *axiomes de l'égalité*. On ne détaillera pas ce système axiomatique ici, voici toutefois quelques indications. Les deux (schémas d')axiomes et la règle de déduction ci-dessous suffisent (à ajouter à ceux du § 2.3):

- 1. $(\forall x\, A(x)) \Rightarrow A(t)$
    où $A(x)$ est une formule, x un symbole de variable, et t un terme.
    t doit être *libre pour x dans $A(x)$*. Cela signifie qu'on peut substituer sans problème t à x dans $A(x)$, le résultat de cette substitution étant une nouvelle formule notée $A(t)$. Exemples: le terme z est libre pour x dans $x \neq y$ et d'ailleurs aussi dans $\exists x\, (x \neq y)$, même si cela présente peu d'intérêt ici (il serait préférable de renommer la variable x liée par le quantificateur existentiel ici, pour obtenir par exemple la formule équivalente $\exists z\, (z \neq y)$). Contre-exemples: le terme a n'est pas libre pour x dans $x \neq a$, ni dans $\exists x\, (x \neq a)$).
  2. $(\forall x\, (A \Rightarrow B)) \Rightarrow (\forall x\, A \Rightarrow \forall x\, B)$ où A et B sont deux formules.
- $\dfrac{A}{\forall x\, A}$ (Généralisation) A est une formule.
- $\exists x\, A$ est par définition $\neg \forall x\, \neg A$ où A est une formule.

Remarque On ne cherche si φ se déduit de $\mathcal{T}$, noté $\mathcal{T} \vdash \varphi$, par cette méthode de *déduction syntaxique* (qui est celle du § 2.3 avec ces deux axiomes et cette règle en plus) que si toutes les formules de $\mathcal{T} \cup \{\varphi\}$ sont closes (comme en fin de § 3.2 pour la conséquence sémantique).

L'égalité $=$ est un symbole de prédicat d'arité 2, qui peut ou non figurer dans un langage $\mathbb{L}$ donné. Mais ce prédicat n'est pas comme les autres, et on distingue les théories avec et sans (les plus simples) égalité. En effet, si ce symbole existe dans $\mathbb{L}$, il faut ajouter des axiomes:

Voici les (schémas d')*axiomes de l'égalité*:

$$\text{(EA)} \quad \forall x \forall y \forall z \left[\begin{array}{l} x=x;\; x=y \Rightarrow y=x;\; x=y \wedge y=z \Rightarrow x=z. \\ x=y \;\Rightarrow\; (P(x) \Leftrightarrow P(y)); \quad x=y \;\Rightarrow\; f(x)=f(y). \end{array} \right. \qquad \text{[(EA) pour “Equality Axiom”]}$$

Les trois premiers axiomes expriment que l'égalité est une *relation d'équivalence* (réflexive, symétrique et transitive). Les axiomes suivants sont appelés les *axiomes de remplacement* pour les prédicats et les fonctions. Il s'agit en fait de schémas d'axiomes: il y a un axiome pour chaque symbole de prédicat P et pour chaque symbole de fonction f du langage $\mathbb{L}$. Il faut aussi considérer tous les axiomes possibles avec tous les symboles d'arité supérieure à 1 existant dans $\mathbb{L}$: $(x_1,x_2,\cdots,x_n) = (y_1,y_2,\cdots,y_n) \;\Rightarrow\; (P(x_1,x_2,\cdots,x_n) \Leftrightarrow P(x_1,x_2,\cdots,x_n))$ et $(x_1,x_2,\cdots,x_n) = (y_1,y_2,\cdots,y_n) \;\Rightarrow\; f(x_1,x_2,\cdots,x_n) = f(x_1,x_2,\cdots,x_n)$, préfixés par $\forall x_1 \forall x_2 \ldots \forall x_n$. Remarquons que ces axiomes sont légèrement redondants: la symétrie et la transitivité sont conséquences de la réflexivité et de l'axiome de remplacement appliqué au prédicat d'égalité $=$.

On a le même **théorème de validité et de complétude** que pour la logique propositionnelle.

Ce résultat est vrai, que $\mathbb{L}$ contienne ou non l'égalité. Si $\mathbb{L}$ contient l'égalité, il faut bien sûr ajouter (EA) aux axiomes donnés en § 2.3 et au début du présent § 3.3, page 7

Par contre, il y a une différence importante. La logique propositionnelle est *décidable*: il existe une procédure de preuve “automatisable” qui, étant donnée une théorie propositionnelle $\mathcal{T}$ et une formule φ répond en un temps fini à la question de savoir si $\mathcal{T} \vdash \varphi$ (c'à-d $\mathcal{T} \models \varphi$) ou non.

La logique du premier ordre n'est plus que *semi-décidable*: il existe une procédure de preuve “automatisable” qui, si $\mathcal{T} \vdash \varphi$ (c'à-d $\mathcal{T} \models \varphi$), permet de le savoir en un temps fini. Mais, si $\mathcal{T} \not\vdash \varphi$, on peut ne jamais obtenir de réponse.

Il existe toutefois des fragments de la logique du premier ordre qui sont décidables. C'est le cas de la logique propositionnelle. C'est aussi le cas de la logique *monadique*, qui ne contient que des symboles de constantes sans aucun autre symbole de fonction, et que des symboles de prédicats d'arité 0 ou 1 (elle peut toutefois contenir le symbole d'égalité). Dès qu'on admet un seul symbole de prédicat binaire (différent de l'égalité), ce n'est plus décidable si on ne fait pas d'autre restriction, même si on n'admet pas le symbole d'égalité.

Notons aussi, parmi de nombreux résultats de ce genre:

Si A est une formule d'un langage $\mathbb{L}$ sans égalité, et si A ne contient ni quantificateur ni symbole de fonction (y compris les constantes), la formule close B égale à $\forall x_1 \cdots \forall x_n\; \exists y_1 \cdots \exists y_m\, A$ est valide ssi elle est vraie dans toute interprétation ayant un domaine fini de n objets ($n \geq 1$). Pour toute formule C de $\mathbb{L}$, il existe une procédure effective permettant de déterminer si C est vraie dans toute interprétation dont le domaine a un nombre fini fixé k d'éléments. Les formules telles que B sont donc décidables pour tout $n \in \mathbb{N}$.

Notons pour terminer qu'il existe des circonstances où on ajoute aussi les axiomes de *séparation des constantes*, ou *d'unicité des noms*: deux termes sont égaux ssi ils s'écrivent exactement de la même manière. Il s'agit d'une condition très forte, qui est parfois utile (programmation logique, ...). Voici ces schémas d'axiomes:

$$\text{(SA)} \qquad \forall x \forall y \left[\begin{array}{l} c_1 \neq c_2; \cdots \ f(x) \neq g(y); \\ f(x) = f(y) \Rightarrow x = y; \cdots, t(x) \neq x \end{array} \right. \qquad \text{[(SA) pour “Separation Axiom”]}$$

$t(x)$ désigne un terme quelconque du langage $\mathbb{L}$, contenant x et différent de x, $\{c_1, c_2, \cdots\}$ est l'ensemble de tous les symboles de constantes du langage $\mathbb{L}$, et f est un symbole quelconque de fonction de $\mathbb{L}$ (comme pour (EA), f peut donc être d'arité quelconque).

La logique des *clauses à variables prédéfinies* est également décidables. Elle non plus ne comporte aucun symbole de fonction d'arité supérieure ou égale à un, et elle n'admet comme formules que les clauses du genre $\neg P(x,y) \vee \neg Q(z) \vee R(x,y,z) \vee T(u)$ dans lesquelles toute variable qui apparaît dans un littéral positif apparaît aussi dans un littéral négatif. Ces clause seront comme toujours en ce cas (implicitement) fermées universellement. Pas d'égalité là.

3.4 Skolémisation (juste quelques indications)

Toute fomule φ peut-être mise sous une forme *prénexe* qui lui est équivalent: tous les quantificateurs en tête, suivis d'une formule sans quantificateur, laquelle peut donc si on veut être mise sous *forme normale conjonctive*, c'est-à dire conjonction de disjonctions d'atomes, équivalent à un ensemble de clauses (*forme clausale*).

Mais il est souvent utile d'aller plus loin, et d'ôter les quantificateurs existentiels, c'est la *skolémisation*. Sans donner la définition technique rigoureuse, voici la description informelle de la méthode, et un exemple. Chaque $\exists x$ va donner naissance à un nouveau symbole de fonction (qui ne figure pas dans le langage initial, et qui n'a pas déjà été employé). Intuitivement, les deux formules “veulent dire un peu la même chose”. Formellement, φ **est satisfiable ssi sa forme skolémisée l'est**.

Voici un exemple:

$\varphi = \forall x \exists y [\exists z \forall w\, R(x,y,z,w) \Rightarrow \exists w\, P(w)]$.

$$\begin{array}{rcl} \text{Forme prénexe: } \varphi & \equiv & \forall x \exists y [\neg \exists z \forall w\, R(x,y,z,w) \vee \exists t\, P(t)] \\ & \equiv & \forall x \exists y [\exists t\, P(t) \vee \forall z \neg \forall w\, R(x,y,z,w)] \\ & \equiv & \forall x \exists y [\exists t\, P(t) \vee \forall z \exists w\, \neg R(x,y,z,w)] \\ & \equiv & \forall x \exists y \exists t \forall z \exists w\, [P(t) \vee \neg R(x,y,z,w)] \end{array}$$

soit, si on préfère: $\forall x \exists y \exists t \forall z \exists w\, [R(x,y,z,w) \Rightarrow P(t)]$.

Skolémisation: Chaque variable quantifiée existentiellement donne naissance à un nouveau symbole de fonction ayant pour arité le nombre de variables quantifiées universellement qui la précède. Intuitivement, ici y et t dépendent de x, et w dépend de x et de z. (On voit sur l'exemple que de nombreuses possibilités existent: si on n'avait pas interverti les termes de la disjonction, on aurait pu obtenir une forme prénexe en $\forall x \exists y \forall z \exists w \exists t$ manifestement inutilement compliquée, puisque t aurait elle aussi donné naissance à un symbole d'arité 2.)

Une forme skolémisée de φ est donc

$\varphi_{sk} = \forall x \forall z [R(x, F_1(x), z, F_3(x,z)) \Rightarrow P(F_2(x))]$,

qui est la clôture universelle de la clause $[\neg R(x, F_1(x), z, F_3(x,z)) \vee P(F_2(x))]$.

On a le résultat suivant: φ est satisfiable ssi φ_{sk} l'est.

(Le langage a changé, ici il faut ajouter trois symboles de fonction.)

Ainsi, *une formule ψ est valide (est une tautologie) ssi la négation d'une forme skolémisée de sa négation l'est*:

$\models \psi$ ssi $\models \neg(\neg\psi)_{sk}$.

Les formes skolémisée permettent donc de ne travailler qu'avec des formules universellement closes, exprimables sous forme clausale, ce qui facilite l'automatisation des calculs (même si la skolémisation elle-même n'est pas gratuite).

3.5 Quelques formules valides (ou tautologies) utiles

φ et ψ sont des formules quelconques,
Φ est ici une formule telle que $x \notin Var(\Phi)$: x ne figure pas comme variable libre dans Φ. Informellement, notons que le $\forall$, qui est une "conjonction généralisée", se comporte bien avec le $\wedge$, tandis que le $\exists$, qui est une "disjonction généralisée", se comporte bien avec le $\vee$.

- $$\begin{array}{rcl} (\forall x\,(\varphi\wedge\psi)) & \Leftrightarrow & (\forall x\,\varphi\wedge\forall x\,\psi) \\ (\exists x\,(\varphi\vee\psi)) & \Leftrightarrow & (\exists x\,\varphi\vee\exists x\,\psi) \\ (\forall x\,(\varphi\Rightarrow\psi)) & \Rightarrow & (\forall x\,\varphi\Rightarrow\forall x\,\psi) \\ (\forall x\,\varphi\wedge\exists x\,\psi) & \Rightarrow & \exists x\,(\varphi\wedge\psi) \end{array} \quad \begin{array}{c} \text{mais seulement} \\ \text{mais seulement} \\ \text{et aussi} \\ \ \end{array} \quad \begin{array}{rcl} (\forall x\,\varphi\vee\forall x\,\psi) & \Rightarrow & (\forall x\,(\varphi\vee\psi)) \\ (\exists x\,(\varphi\wedge\psi)) & \Rightarrow & (\exists x\,\varphi\wedge\exists x\,\psi) \\ (\forall x\,(\varphi\Rightarrow\psi)) & \Rightarrow & (\exists x\,\varphi\Rightarrow\exists x\,\psi) \\ & & \end{array}$$

- $$\begin{array}{rcl} (\forall x\,(\Phi\wedge\psi)) & \Leftrightarrow & (\Phi\wedge\forall x\,\psi) \\ (\forall x\,(\Phi\Rightarrow\psi)) & \Leftrightarrow & (\Phi\Rightarrow\forall x\,\psi) \\ (\exists x\,(\Phi\wedge\psi)) & \Leftrightarrow & (\Phi\wedge\exists x\,\psi) \\ (\exists x\,(\Phi\Rightarrow\psi)) & \Leftrightarrow & (\Phi\Rightarrow\exists x\,\psi) \end{array} \quad \begin{array}{c} \ \\ \textbf{et, attention:} \\ \ \\ \text{et, attention:} \end{array} \quad \begin{array}{rcl} (\forall x\,(\Phi\vee\psi)) & \Leftrightarrow & (\Phi\vee\forall x\,\psi) \\ (\forall x\,(\psi\Rightarrow\Phi)) & \Leftrightarrow & ((\exists x\,\psi)\Rightarrow\Phi) \\ (\exists x\,(\Phi\vee\psi)) & \Leftrightarrow & (\Phi\vee\exists x\,\psi) \\ (\exists x\,(\psi\Rightarrow\Phi)) & \Leftrightarrow & ((\forall x\,\psi)\Rightarrow\Phi) \end{array}$$

- $\forall x\,\varphi \quad \Rightarrow \quad \exists x\,\varphi$

- $$\begin{array}{rcl} (\forall x\forall y\ \ x=y) & \Leftrightarrow & (\exists x\forall y\ \ x=y) \\ (\forall x\forall y\forall z\,(x=y\vee x=z\vee y=z)) & \Leftrightarrow & (\exists x\exists y\forall z\,(z=x\vee z=y))) \end{array}$$

$\varphi[x]$ est une formule quelconque, $t, t_1, \cdots t_n$ sont des termes ne contenant ni x ni variable liée dans $\varphi[x]$; $\varphi[t]$ désigne la formule obtenue en remplaçant chaque occurrence libre de x dans $\varphi[x]$ par le terme t.

- $\varphi[t] \quad \Leftrightarrow \quad (\forall x\,(x=t\ \Rightarrow\ \varphi[x])) \qquad$ et $\qquad \varphi[t] \quad \Leftrightarrow \quad (\exists x\,(x=t\ \wedge\ \varphi[x]))$

  $$\begin{array}{rcl} (\varphi[t_1]\wedge\cdots\wedge\varphi[t_n]) & \Leftrightarrow & (\forall x\,((x=t_1\vee\cdots\vee x=t_n)\ \Rightarrow\ \varphi[x])) \\ (\varphi[t_1]\vee\cdots\vee\varphi[t_n]) & \Leftrightarrow & (\exists x\,((x=t_1\vee\cdots\vee x=t_n)\ \wedge\ \varphi[x])) \end{array}$$

3.6 Logiques du premier ordre typées

Il y a des *types* d'objets, de fonctions et de prédicats. Sans entrer dans les détails formels, par exemple pour le calcul vectoriel il y aura deux types d'objets, les *scalaires* et les *vecteurs*.

On peut toujours plonger une telle logique dans une logique non typée en introduisant des symboles de prédicats. Dans notre exemple, on introduirait S et V avec $S(s)$ vrai ssi s est un scalaire et $V(v)$ vrai ssi v est un vecteur. Si donc on note respectivement en $s, s_{...}$ et $v, v_{...}$ les variables de type scalaire et celles de type vecteur, on passe de l'écriture typée à l'écriture non typée ainsi (décrit informellement):

On remplace tous les $\forall s\ \Phi(s)$, où $\Phi(s)$ est une formule, par $\forall s\ (S(s) \Rightarrow \Phi(s))$, et tous les $\exists s\ \Phi(s)$, par $\exists s\ (S(s) \wedge \Phi(s))$, et de même pour V au lieu de S.

Ainsi, on peut dire que ces logiques n'apportent rien aux logiques non typées. Mais en fait, utiliser des logiques typées permet, et c'est important, une grande simplification de l'écriture, et aussi parfois une grande simplification de l'automatisation des calculs de déduction.

4 Quelques extensions non classiques des logiques classiques

4.1 Minimisation des informations positives

Toutes les extensions étudiées en ce § 4 consistent en quelque sorte à minimiser des informations positives. Ces extension ont toutes été initiées par l'informatique et le besoin de traduire et traiter des situations réelles. Il existe bien d'autres extensions, en plus des logiques modales évoquées en § 5, non abordées dans ce texte: logiques à plus de deux valeurs de vérité, logiques para-consistantes où on peut avoir A et $\neg A$ sans pour autant pouvoir tout déduire, etc...

Les problèmes qui ont conduit aux logiques traitées dans ce §4:
- informations incomplètes
- règles avec exceptions

Informations incomplètes:
On ne peut en général pas expliciter tout ce qui est faux (base de données, ...) Ex. Horaires d'avion.

On précise (directement ou non) toutes les *informations positives*, les *informations négatives* auront un sort particulier.

Exemple: On sait: D = $\{$*Cube(a), Cylindre(b), Cube(c), Posé-sur(a,b)*$\}$

Questions	*Cylindre(a)*?	Réponse	NON,
	Cube(b)?		NON
	Posé-sur(a,c)?		NON, etc...

Règles avec exceptions: *les oiseaux volent*. Ce problème peut être traité par les mêmes méthodes que le problème précédant: on peut le traduire par *les oiseaux non exceptionnels volent*, et chercher à *minimiser les exceptions*.

4.2 Hypothèse du monde clos(HMC), ou Closed World Assumption (CWA)

Définition 4.1 Soit une base de données exprimée grâce à un ensemble de formules de logique D. On ajoute à D l'ensemble de tous les littéraux négatifs concrets (= instanciés) tels que le littéral complémentaire n'est pas conséquence logique de D. Cela forme $HMC(D) = Th(D \cup D^{HMC})$.

Donc, pour tout littéral positif concret $A(t)$, on a $\neg A(t) \in D^{HMC}$ ssi $A(t) \notin Th(D)$.

L'hypothèse du monde clos permet une économie de représentation, mais ne fait pas de distinction entre information fausse et information inconnue.

Si D ne contient que des littéraux, il n'y a pas de problème, mais pour les formules quelconques, la consistance n'est pas garantie:

D = $\{\forall x\,(Individu(x) \Rightarrow (Homme(x) \vee Femme(x))),\ $ *Individu(Dominique)*$\}$

$HMC(\mathrm{D}) \vdash \neg$*Homme(Dominique)*, $HMC(\mathrm{D}) \vdash \neg$*Femme(Dominique).*

Il s'agit d'un ensemble de clauses (cf fin du § 2.3 page 4).

Le problème vient de ce qu'on doit avoir (même sous-jacente, c'à-d même si elle n'est pas totalement exprimée) une *connaissance complète* du domaine considéré. Les disjonctions provoquent l'inconsistance:

Propriété 4.1 Si D ne contient que des *clauses de Horn* (disjonctions de littéraux avec au plus un littéral positif), et est consistant, alors $HMC(\mathrm{D})$ est consistant.

HMC dépend du vocabulaire: qu'est-ce qu' une *information positive*?

4.3 Complétion de prédicat

Même idée: pour les informations positives, seulement ce qui **doit** être vrai est vrai.

Les *si* des définitions sont traités comme des *si et seulement si* implicites.

Le même exemple: D = $\{$Cube*(a)*, Cylindre*(b)*, Cube*(c)*, Posé-sur*(a,b)*$\}$.

On peut ré-écrire ces données sous la forme (formules devant être fermées universellement, c'est-à-dire que l'on sous-entend la clôture par $\forall$):

$$\begin{array}{lrcl} \mathrm{D} \;=\; & \{x=a & \Rightarrow & Cube(x), \\ & x=b & \Rightarrow & Cylindre(x), \\ & x=c & \Rightarrow & Cube(x), \\ & x=a \wedge y=b & \Rightarrow & Posé\text{-}sur(x,y)\}, \quad \text{c'à-d} \end{array}$$

$$\begin{array}{lrcl} \mathrm{D} \;=\; & \{x=a \vee x=c & \Rightarrow & Cube(x), \\ & x=b & \Rightarrow & Cylindre(x), \\ & x=a \wedge y=b & \Rightarrow & Posé\text{-}sur(x,y)\}. \end{array}$$

La base de donnée complétée est

$$\begin{array}{rcrcl} \mathrm{D}^c & = & \{x=a\vee x=c & \Leftrightarrow & Cube(x), \\ & & x=b & \Leftrightarrow & Cylindre(x), \\ & & x=a\wedge y=b & \Leftrightarrow & Posé\text{-}sur(x,y)\}. \end{array}$$

Sur cet exemple, le résultat est le même qu'avec HMC, à condition toutefois d'ajouter quelques axiomes logiques: (EA), (SA) et aussi l' *axiome de fermeture du domaine* (DCA):

Si $\mathcal{T}$ est un ensemble de formules dans un langage logique $\mathbb{L}$ sans symbole de fonction et ne contenant qu'un nombre fini de symboles de constantes $c_1, c_2, \cdots$

(DCA) $\quad \forall x\,(x=c_1 \vee x=c_2 \vee \cdots)$.

(si les conditions ne sont pas remplies, on ajoute parfois cet "axiome" quand même, mais ce n'est plus une formule de la logique du premier ordre).

On trouve là encore: $\mathrm{D} \vdash Cube(a)$, etc... donc $\mathrm{D}^c \vdash Cube(a)$, etc..,

mais aussi, comme avec HMC(D): $\mathrm{D}^c \vdash \neg Cube(b)$, $\mathrm{D}^c \vdash \neg Cylindre(a)$, etc...

Définition 4.2 $\mathcal{T}$ est un ensemble de clauses. Toute clause dont la *tête* est en P,

$P(t_1,\cdots,t_n) \Leftarrow L_1 \wedge \cdots \wedge L_m,$ ayant $\{y_1,\cdots,y_p\}$ pour ensemble de variables libres, équivaut à (cf § 3.5):

$P(x_1,\cdots,x_n) \Leftarrow \exists y_1\cdots y_p\;(x_1=t_1 \wedge x_n=t_n \wedge L_1 \wedge \cdots \wedge L_m)$, notée ici

$P(x_1,\cdots,x_n) \Leftarrow E_i$.

Supposons qu'on ait j telles clauses au départ, alors leur réunion, appelée la *définition de P dans* $\mathcal{T}$, équivaut à: $\quad P(x_1,\cdots,x_n) \Leftarrow E_1 \vee \cdots \vee E_j$.

C'est la *forme générale du prédicat P dans* $\mathcal{T}$.

La *complétion du prédicat P dans* $\mathcal{T}$ consiste à remplacer l'implication par une équivalence logique dans la forme générale de P dans $\mathcal{T}$: $\quad P(x_1,\cdots,x_n) \Leftrightarrow E_1 \vee \cdots \vee E_j$.

On ajoute aussi (EA), (SA) et, en général, (DCA).

Exemple 4.1 $\mathcal{T}= \{Vole(x) \Leftarrow Oiseau(x), Vole(Icare), Oiseau(Titi), \neg Oiseau(Mirza)\}$.

COMP($\mathcal{T}$*:* Vole*)* ajoute à $\mathcal{T}$ la formule suivante: $Vole(x) \Leftrightarrow (Oiseau(x) \vee x = Icare)$.

COMP($\mathcal{T}$*:* Oiseau*)* ajoute à $\mathcal{T}$ la formule suivante: $Oiseau(x) \Leftrightarrow x = Titi$.

COMP($\mathcal{T}$*:* Vole, Oiseau*)* ajoute ces deux formules à $\mathcal{T}$.

La complétion (de l'ensemble de tous les prédicats) et HMC se ressemblent, mais ont parfois un comportement différent: la complétion dépend de la syntaxe alors que HMC ne dépend que du vocabulaire choisi. En cas de clauses de Horn, la complétion et l'hypothèse du monde clos sont chacune consistantes, mais pas toujours équivalentes (même en ajoutant (EA), (SA) et (DCA)).

Remarque: La complétion a un comportement voisin de l'opérateur de négation de PROLOG (HMC aussi, d'ailleurs, mais "moins proche"). Sous certaines restrictions, il y a même une très bonne adéquation entre complétion de tous les prédicats et "non" (ou "not") de PROLOG.

4.4 La circonscription de prédicats

Créée pour remédier aux inconvénients de HMC et de la complétion. Indépendante de la syntaxe ($\neq$ complétion), plus souvent consistante que HMC. Même idée de départ: n'est vrai que ce que les données contraignent à considérer vrai (en ce qui concerne certaines informations positives).

Définitions 4.3 $\mathcal{T}[\mathbf{P},\mathbf{Q}]$ est un ensemble fini de formules, assimilé à la conjonction de ses éléments, dans un langage de la logique du premier ordre $\mathbb{L}$, contenant entre autre les symboles de prédicats de $\mathbf{P} = (P_1, P_2, \cdots, P_n)$ et de $\mathbf{Q} = (Q_1, Q_2, \cdots, Q_m)$.

La *circonscription de* **P** *dans* $\mathcal{T}$, *en laissant* **Q** *varier*, notée $Circ(\mathcal{T} : \mathbf{P};\ \mathbf{Q})$, consiste à ne garder que les modèles de $\mathcal{T}$ minimaux pour la relation $<_{\mathbf{P};\mathbf{Q}}$ définie ci-dessous.

μ et ν désignant des interprétations dans $\mathbb{L}$, on définit $<_{\mathbf{P};\mathbf{Q}}$ par: $\mu <_{\mathbf{P};\mathbf{Q}} \nu$ si:

- L'extension de chaque P_i dans μ est un sous-ensemble de l'extension de P_i dans ν:
  $|P_i|_\mu \subseteq |P_i|_\nu$ $(1 \leq i \leq n)$, avec inclusion stricte $|P_i|_\mu \subset |P_i|_\nu$ pour au moins un des i.
- Les deux interprétations sont absolument identiques par ailleurs...
- ... sauf qu'il n'y a aucune condition sur les extensions des Q_i.

Un modèle μ de $\mathcal{T}$ est *minimal pour* $<_{\mathbf{P};\mathbf{Q}}$ s'il n'existe aucun modèle ν de $\mathcal{T}$ tel que: $\nu <_{\mathbf{P};\mathbf{Q}} \mu$.

Il existe un schéma d'axiomes. Notons qu'il faut ajouter (EA) (on doit avoir l'égalité) et qu'en général on ajoute aussi (SA), ou au moins une partie de (SA).

Définition 4.4 $\mathcal{T}[P,Q]$ est un ensemble fini de formules, assimilé à la conjonction de ses éléments, dans un langage de la logique du premier ordre $\mathbb{L}$, contenant entre autre les symboles de prédicats P et Q.

Le schéma de *circonscription du prédicat P dans $\mathcal{T}$ en laissant le prédicat Q varier* s'écrit:

$$\text{(SAC)}\ \ (\mathcal{T}[p,q] \wedge \forall \mathbf{x}\,(p[\mathbf{x}] \Rightarrow P(\mathbf{x}))) \ \Rightarrow\ \forall \mathbf{x}\,(P(\mathbf{x}) \Rightarrow p[\mathbf{x}]).$$

La *circonscription de P dans $\mathcal{T}$, en laissant Q varier*, notée

$$Circ(\mathcal{T} : P;\ Q),$$

ajoute à $\mathcal{T}$= $\mathcal{T}$[P,Q], toutes les instances du schéma d'axiomes (SAC), pour toutes les formules p et q de $\mathbb{L}$ possibles.

• $\mathbf{x}$ désigne une suite finie de variables $x_1,\cdots,x_n$, où n est l'arité de P.

• p, notée aussi $p[\mathbf{x}]$, ainsi que q, désignent des formule de $\mathbb{L}$ quelconques.

• $\mathcal{T}[p,q]$ désigne la formule $\mathcal{T}[P,Q]$ dans laquelle chaque occurrence du prédicat P est remplacée par l'occurrence correspondante de p, et chaque occurrence du prédicat Q est remplacée par l'occurrence correspondante de q ($P(t) \rightsquigarrow p[t]$, et de même pour Q).

• P et Q peuvent aussi désigner des suites finies de prédicats:

Si $P = (P_1,P_2)$, alors $\forall \mathbf{x}\ \ (p[\mathbf{x}] \Rightarrow P(\mathbf{x}))$ est une abréviation pour $\forall x_1x_2y_1y_2y_3\ ((p_1[x_1,x_2] \Rightarrow P_1(x_1,x_2)) \wedge (p_2[y_1,y_2,y_3] \Rightarrow P_2(y_1,y_2,y_3)))$ où P_1 et P_2 sont deux symboles de prédicats, d'arités respectives 2 et 3, et où $p_1[x_1,x_2]$ et $p_2[y_1,y_2,y_3]$ sont des formules quelconques de $\mathbb{L}$. La grosse difficulté est de trouver les bonnes instances du schéma de circonscription. Les systèmes de calcul n'utilisent pas directement cette définition.

Exemple 4.2 $\mathcal{T}$= {*Cube(a), Cylindre(b), Cube(c), Posé-sur(a,b)*}.

$Circ(\mathcal{T} : Cube)$ ajoute à $\mathcal{T}$ la formule $Cube(x) \Leftrightarrow (x = a \vee x = c)$ (cf complétion). Si donc on ajoute (SA), on déduit $\neg Cube(b)$.

Circ($\mathcal{T}$: *Posé-sur)* ajoute à $\mathcal{T}$ la formule *Posé-sur*$(x,y) \Leftrightarrow (x = a \wedge y = b)$
(toujours: clôture universelle, par $\forall\cdots$, des formules, ici donc $\forall x \forall y\cdots$).

Circ($\mathcal{T}$: *Cube, Cylindre, Posé-sur*) ajoute à $\mathcal{T}$ les formules
$Cube(x) \Leftrightarrow (x = a \vee x = c)$, $Cylindre(x) \Leftrightarrow (x = b)$ et *Posé-sur*$(x,y) \Leftrightarrow (x = a \wedge y = b)$.

Ajoutons maintenant un autre symbole de prédicat au langage $\mathbb{L}$ de $\mathcal{T}$: Boule (mais sans ajouter de formule).
$Circ(\mathcal{T} : \text{Boule}) =$?
(pas de boule: $\forall x\ \neg Boule(x)$ soit $Boule(x) \Leftrightarrow \bot$, l'instance associée correspond au choix suivant pour la formule ($p[x]$ de la définition du schéma d'axiomes de circonsription) $boule[x] = \bot$).

Bien sûr, si une instance du schéma de circonscription de $Circ(\mathcal{T} : P;Q)$ permet de démontrer $\forall x\ \neg P(x)$, alors $Circ(\mathcal{T} : P;Q) \equiv \mathcal{T}\ \wedge\ \forall x\ \neg P(x)$ (on ne peut pas espérer minimiser davantage p!).

Sur cet exemple des solides: circonscription de $P\ \ =\ \ $complétion de P.

Plus généralement, on a les résultats suivants:

- Si $\mathcal{T}$ est une théorie de Horn, alors la circonscription $Circ(\mathcal{T}:P)$ implique la complétion de P dans $\mathcal{T}$.

- Si $\mathcal{T}$ est une théorie de Horn, sans *clause récursive en P* (c'est-à dire que P ne figure jamais en tête et en queue d'une même clause), alors $Circ(\mathcal{T}:P)$ équivaut à la complétion de P dans $\mathcal{T}$.

Exemple: $\mathcal{T}$: $P(a)\vee P(b)$. $Circ(\mathcal{T}:P)\equiv(\forall x\,(P(x)\Leftrightarrow x=a))\vee(\forall x\,(P(x)\Leftrightarrow x=b))$. Le $\vee$ est devenu "ou exclusif" (si $a\neq b$). Le plus simple est ici de prendre deux instances: celle associée à $p[x]\equiv x=a$ **et** celle associée à $p[x]\equiv x=b$.

Exemple: $\mathcal{T}$: $P(a)\Rightarrow P(b)$. $Cir(\mathcal{T}:P)\equiv\forall x\,\neg P(x)$.

Un cas typique d'utilisation de la circonscription concerne ***la circonscription (ou minimisation) des exceptions***.

Exemple: $\mathcal{T}$: $(Oiseau(x)\wedge\neg Exc(x))\Rightarrow Vole(x)$, $Oiseau(Titi)$, $Oiseau(Zoé)$, $\neg$*Vole(Zoé).*

$Circ(\mathcal{T}:Exc;\,Vole)$ ajoute à $\mathcal{T}$ la formule $Exc(x)\Leftrightarrow x=Zoé$.

L'instance du schéma de circonscription qui permet de démontrer ce résultat est celle qui est associée au choix de formules: $exc[x]\equiv x=Zoé$, $vole[x]\equiv Oiseau(x)\wedge x\neq Zoé$. D'où l'intérêt d'utiliser d'autres méthodes pour l'automatisation...

La sémantique des modèles minimaux (définition 4.3, c'est la sémantique attendue) **ne correspond pas toujours exactement à** $Circ(\mathcal{T}:P;Q)$ telle qu'elle a été définie syntaxiquement en définition 4.4 (*c*irconscription du premier ordre). On a seulement:

Tout modèle de $\mathcal{T}$ minimal pour $(P;Q)$ **est un modèle de la circonscription du premier ordre Circ(**$\mathcal{T}:P;Q$).

Il faut donc parfois introduire la version du second ordre:

$\mathcal{T}$ demeure du premier ordre, et on ne s'intéresse qu'aux conséquences du premier ordre, mais on passe en second ordre

(ce qui signifie qu'on introduit des *v*ariables de prédicats et de fonctions – dans la version donnée ci-dessous les prédicats suffisent – sur lesquelles on pourrait quantifier)

pour définir ***l'*axiome de circonscription (AC)**:

Définition 4.5 $Circ_2(\mathcal{T}:P;Q)=\mathcal{T}\wedge(AC)$, où

(AC) $\equiv\forall p\,q\,((\mathcal{T}[p,q]\wedge\forall\mathbf{x}\,(p(\mathbf{x})\Rightarrow P(\mathbf{x})))\;\Rightarrow\;\forall\mathbf{x}\,(P(\mathbf{x})\Rightarrow p(\mathbf{x})))$.

p et q sont des symboles de variable de prédicats, ayant respectivement la même arité que les symboles de constantes de prédicat P et Q.

On a alors la sémantique attendue:

Un modèle de $\mathcal{T}$ est un modèle de la circonscription du second ordre $Circ_2(\mathcal{T}:P;Q)$ (définition 4.5) **ssi c'est un modèle de $\mathcal{T}$ minimal pour** $(P;Q)$ (cf définitions 4.3).

5 Logiques modales classiques *("pour la culture", pas utilisé dans ce cours)*

5.1 Motivations

La logique du premier ordre est *vérifonctionnelle*: la valeur de vérité d'une formule dépend uniquement des valeurs de vérité de ses sous-formules (ex. $A\wedge B$).

Il existe des situations où ce n'est pas bien adapté. Ex.:

Il pleut, noté P_1, ne donne pas d'indication sur

Je sais qu'il pleut, noté $\mathbf{L}P_1$, ni d'ailleurs sur

Je sais qu'il ne pleut pas, noté $\mathbf{L}\neg P_1$.

Il en est de même si $\mathbf{L}P_1$ veut représenter la phrase:

Je sais qu'il pleuvra un jour dans le futur.

Contrairement aux extensions vues en § 4, l'origine des logiques modales est très ancienne: moyen âge et antiquité (Aristote). Mais les besoins de l'informatique ont provoqué un développement considérable. Formellement, les logiques modales sont caractérisées par des opérateurs modaux, qui viennent en complément des symboles logiques classiques. Il existe des *logique propositionnelles modales* et des *logiques du premier ordre modales*. Un *opérateur modal*, appliqué à une (ou plusieurs) proposition(s), donne une nouvelle proposition.

Exemple d'opérateurs modaux classiques, L peut être:
K: *je sais* (logiques *épistémiques*, ou logiques de la connaissance),
L: *il est nécessaire que* (logiques *alétiques*),
O: *il est obligatoire que* (sens juridique, logiques *déontiques*),
G: *il sera toujours le cas (que)* (logiques *temporelles*).

Exemple d'opérateurs modaux avec deux arguments:

R: P_1 **R** P_2 P_1 *depuis* P_2, depuis l'instant où P_2 a été vrai, P_1.
S: P_1 **S** P_2 P_1 *jusqu'à* P_2, P_1 est (a tj été et sera tj) vrai, jusqu'à ce que P_2 soit vrai.

	$\cdots$	x	x	x	x	x	x	x	x	x	x	x	x	$\cdots$
P_1 **R** P_2:	$\cdots$	P_2	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	$\cdots$
P_1 **S** P_2:	$\cdots$	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_1	P_2	$\cdots$

Ce sont deux autres exemples de logique temporelle (modale).

$\rightarrow$: $A \rightarrow B$: *si A était vérifié, alors B serait vérifié* (logique *conditionnelle*).
Il existe des systèmes *multi-modaux*, où plusieurs opérateurs modaux a priori non reliés entre eux coexistent: *Je sais qu'il ne pleuvra pas*: $\mathbf{K} \neg \mathbf{G}\, P_1$.
Je sais qu'il est possible qu'il pleuve un jour: $\mathbf{K} \neg \mathbf{L}\ \ \mathbf{G} \neg P_1$.

Même pour les logiques qui ne nécessitent fondamentalement qu'un opérateur modal, il est fréquent d'introduire son *dual*: **M** est défini comme étant $\neg\mathbf{L}\neg$.

L A	**M** A
Je sais que A	Je ne sais pas que $\neg A$
Il est nécessaire que A	Il est possible que A
Il est obligatoire que A	Il est permis que A
Il sera toujours vrai que A	Il sera parfois vrai que A
Toute exécution du programme produit A	Il existe une exécution du programme qui produit A

Exemples de formules modales avec variables:
$\exists x\,(\mathbf{L}E(x))$: il existe quelqu'un dont je sais qu'il est un espion, c'est-à-dire: je connais un espion.
$\mathbf{L}(\exists x\, E(x))$: je sais qu'il y a un espion (mais sans savoir qui).

Noter que **L** est souvent noté $\Box$ et alors son dual **M** est noté $\diamond$. En cas de systèmes multi-modaux, on note souvent $[i]$ et $< i >$ deux opérateurs modaux duaux, pour plusieurs valeurs différentes de i. Ainsi, dans la logique temporelle K_t on aura les opérateurs $[P]$ et $[F]$ (*passé et futur "forts"*) ainsi que leurs duaux $<P>$ et $<F>$ (*passé et futur "faibles"*).

5.2 Logiques modales classiques (syntaxe)

Définition 5.1 Soit un langage modal $\mathbb{L}_{\mathbf{L}}$ (un langage du premier ordre $\mathbb{L}$ augmenté d'un opérateur modal unaire **L**).

- Toute formule atomique de $\mathbb{L}$ est une formule atomique de $\mathbb{L}_{\mathbf{L}}$.

- Toute formule atomique de $\mathbb{L}_{\mathbf{L}}$ est une formule (bien formée) de $\mathbb{L}_{\mathbf{L}}$.
- Si φ et ψ sont des formules de $\mathbb{L}_{\mathbf{L}}$, alors $\neg\varphi$, $\varphi \wedge \psi$, $\forall x\varphi$ et $\mathbf{L}\ \varphi$ sont des formules de $\mathbb{L}_{\mathbf{L}}$.
- Les seules formules de $\mathbb{L}_{\mathbf{L}}$ sont les formules obtenues à partir des trois règles ci-dessus.

On définit aussi $\vee$, $\Rightarrow$, $\exists$ et **M** (**M** φ est $\neg\mathbf{L}\neg\varphi$).
On se limitera dans ces pages aux logiques modales propositionnelles.

Définition 6.1: Soit un langage modal $\mathbb{L}_{\mathbf{L}}$ (un langage propositionnel $\mathbb{L}$ augmenté d'un opérateur modal unaire **L**).

- Toute formule atomique de $\mathbb{L}$ est une formule atomique de $\mathbb{L}_{\mathbf{L}}$.
- Toute formule atomique de $\mathbb{L}_{\mathbf{L}}$ est une formule (bien formée) de $\mathbb{L}_{\mathbf{L}}$.
- Si φ et ψ sont des formules de $\mathbb{L}_{\mathbf{L}}$, alors $\neg\varphi$, $\varphi \wedge \psi$, et $\mathbf{L}\ \varphi$ sont des formules de $\mathbb{L}_{\mathbf{L}}$.
- Les seules formules de $\mathbb{L}_{\mathbf{L}}$ sont les formules obtenues à partir des trois règles ci-dessus.

On définit aussi $\vee$, $\Rightarrow$, et **M** (**M** φ est $\neg\mathbf{L}\neg\varphi$).

Définition 5.2 Le système modal K a pour langage $\mathbb{L}_{\mathbf{L}}$, où $\mathbb{L}$ est un langage de logique propositionnelle (pas de prédicat, pas de variables d'individu). Il contient les (schémas d') axiomes suivants:

- Toutes les tautologies du calcul propositionnel [2].
- L'axiome K: $\mathbf{L}(\varphi \Rightarrow \psi) \Rightarrow (\mathbf{L}\varphi \Rightarrow \mathbf{L}\psi)$.

Il contient les règles d'inférences suivantes:

- Modus ponens: si φ et $\varphi \Rightarrow \psi$ sont des théorèmes[3], ψ aussi (comme en logique classique)
- La règle de nécessitation: si φ est un théorème[3], alors $\mathbf{L}\ \varphi$ aussi.

Un système modal qui contient le système K, est appelé un système modal *normal*.

Propriété 5.1 Dans tout théorème d'un système modal normal, on peut remplacer une sous-formule par une sous-formule équivalente.

En particulier on a: Si $\mathbf{L}\varphi$ et $\varphi \Leftrightarrow \varphi'$ sont des théorèmes, alors $\mathbf{L}\varphi'$ est aussi un théorème.

Il existe des systèmes modaux qui ne vérifient pas cette propriété, et il faut alors être très prudent dans les preuves.

Dans les systèmes non normaux, la sémantique des mondes possibles vues ci-dessous n'est plus utilisable aussi directement.

Définition 5.3 Autres schémas d'axiomes classiques:

- L'axiome T: $\mathbf{L}\varphi \Rightarrow \varphi$ (tout ce qui est nécessaire est vrai).
- L'axiome D: $\mathbf{L}\varphi \Rightarrow \mathbf{M}\varphi$ (c'à-d: $\mathbf{L}\varphi \Rightarrow \neg\mathbf{L}\neg\varphi$) (tout ce qui est nécessaire est possible).
- L'axiome 4: $\mathbf{L}\varphi \Rightarrow \mathbf{LL}\varphi$ *auto-introspection positive* (je sais tout ce que je sais).
- L'axiome 5: $\mathbf{M}\varphi \Rightarrow \mathbf{LM}\varphi$ *auto-introspection négative* (équivalent à $\neg\mathbf{L}\varphi \Rightarrow \mathbf{L}\neg\mathbf{L}\varphi$).

Cela fournit les systèmes (propositionnels) modaux appelés KT ou simplement T, $KT4$ ou $S4$, $KT45$ ou $S5$, $KD45$, ...

---

[2] Y compris le *principe de substitution uniforme*: on peut, dans toute tautologie du calcul propositionnel, remplacer uniform´ement tout symbole propositionnel par une formule quelconque de $\mathbb{L}_{\mathbf{L}}$.

[3] th´eor`eme du syst`eme consid´er´e (ici K).

Notons que, par exemple, le schéma d'axiome 4 équivaut au schéma d'axiomes $\mathbf{MM}\varphi \Rightarrow \mathbf{M}\varphi$ (remplacer dans l'axiome 4, φ par $\neg\varphi$, sachant que "pour toute formule φ du langage" équivaut à "pour toute formule ${}_{n}on\varphi$ du langage").

Remarque: Le théorème de la déduction n'est pas vrai en logique modale. En particulier, ne pas confondre la formule $P \Rightarrow \mathbf{L}P$ et la règle de nécessitation! La formule donnée ici n'est certainement pas valide dans KT, ni dans aucun système modal utile contenant T.

Propriété 5.2 *Quelques tautologies (formules vraies) du système KT:*

1. $\varphi \Rightarrow \mathbf{M}\varphi$
2. $\mathbf{L}(\varphi \wedge \psi) \Leftrightarrow (\mathbf{L}\varphi \wedge \mathbf{L}\psi)$
3. $\mathbf{M}(\varphi \vee \psi) \Leftrightarrow (\mathbf{M}\varphi \vee \mathbf{M}\psi)$
4. $(\mathbf{L}\varphi \vee \mathbf{L}\psi) \Rightarrow \mathbf{L}(\varphi \vee \psi)$
5. $\mathbf{M}(\varphi \wedge \psi) \Rightarrow (\mathbf{M}\varphi \wedge \mathbf{M}\psi)$
6. $\mathbf{L}\varphi \Rightarrow \mathbf{M}\varphi$ *(axiome D).*

Remarque: L'axiome K suffit pour les formules 2 à 5. L'axiome T est inutile là, ce qui est important si on considère une logique qui ne satisfait pas T, comme par exemple la logique temporelle K_t (à ne pas confondre avec KT!). Ainsi, dans K_t, on aura les formules (notations "françaises"):
$[P](\varphi \wedge \psi) \Leftrightarrow ([P]\varphi \wedge [P]\psi)$ et $[F](\varphi \wedge \psi) \Leftrightarrow ([F]\varphi \wedge [F]\psi)$, mais ni $\varphi \Rightarrow <P> \varphi$ ni $[F]\varphi \Rightarrow <F> \varphi$.

5.3 Sémantique: univers des mondes possibles de Kripke

Définition 5.4 Une *structure modale* est un quadruplet (W, w_0, I, R) où:

- W est un ensemble non vide dont les éléments w sont les *mondes*;
- w_0 est un élément de W (*le monde réel*);
- R est une relation binaire sur W. Si wRw' on dit que w' est *accessible* à partir de w;
- I associe une interprétation propositionnelle à chaque élément de W. Ainsi, à chaque $w \in W$, I associe un ensemble de symboles propositionnels du langage $\mathbb{L}$ considéré. Si $P \in I(w)$, on dit que P est vrai dans le monde w pour cette structure, sinon P est faux dans le monde w pour cette structure.

Définition 5.5 La notion de vérité, pour toute formule de $\mathbb{L}_{\mathbf{L}}$, dans les mondes w de W est établie ainsi:

- $w \models P$ ssi $P \in I(w)$, pour chaque symbole propositionnel de $\mathbb{L}$;
- $w \models \neg\varphi$ ssi $w \not\models \varphi$;
- $w \models \varphi \wedge \psi$ ssi $w \models \varphi$ et $w \models \psi$;
- $w \models \mathbf{L}\varphi$ ssi $w' \models \varphi$ pour tout monde $w' \in W$ tel que wRw'.

Une formule φ de $\mathbb{L}_{\mathbf{L}}$ est *valide dans une structure modale* (W, w_0, I, R) ssi $w_0 \models \varphi$.
On note alors: $(W, w_0, I, R) \models \varphi$).
Une formule φ de $\mathbb{L}_{\mathbf{L}}$ est *valide* (ou est une tautologie) ssi elle est valide dans toute structure modale (on note $\models \varphi$).

Théorème 5.3 Validité et complétude des principaux systèmes modaux:

- Une formule φ de $\mathbb{L}_{\mathbf{L}}$ est un théorème de KT ssi φ est valide dans toutes les structures modales dont la relation R est réflexive.
- Une formule φ est un théorème de $KT4$ (aussi appelé $S4$), ssi φ est valide dans toutes les structures modales dont la relation R est réflexive et transitive.
- Une formule φ est un théorème de $KT45$ (aussi appelé $S5$), ssi φ est valide dans toutes les structures modales dont la relation R est une relation d'équivalence. On a d'ailleurs aussi:

Une formule φ est un théorème de $S5$ ssi φ est valide dans toutes les structures modales dont la relation R est la relation universelle (partout vraie).

- Une formule φ est un théorème de $KD45$ ssi φ est valide dans toutes les structures modales dont la relation R est euclidienne, reproductive et transitive.

On a une correspondance individuelle entre les axiomes modaux classiques déjà donnés et les propriétés que doit satisfaire la relation d'accessibilité R:

- L'axiome T: $\mathbf{L}\varphi \Rightarrow \varphi$ correspond à une relation *réflexive*
- L'axiome D: $\mathbf{L}\varphi \Rightarrow \mathbf{M}\varphi$ correspond à une relation *reproductive*.
- L'axiome 4: $\mathbf{L}\varphi \Rightarrow \mathbf{LL}\varphi$ correspond à une relation *transitive*
- L'axiome 5: $\mathbf{M}\varphi \Rightarrow \mathbf{LM}\varphi$ correspond à une relation *euclidienne*
- L'axiome $\varphi \Rightarrow \mathbf{LM}\varphi$ correspond à une relation symétrique (cet axiome est une conséquence de T et (5)).

Définition 5.6 Une relation *euclidienne* vérifie:

$$\forall w_1 \forall w_2 \forall w_3\ ((w_1Rw_2 \wedge w_1Rw_3) \Rightarrow w_2Rw_3).$$

Une relation *reproductive* vérifie: $\forall w_1 \exists w_2\ (w_1Rw_2)$.

Remarque: • Une relation symétrique et transitive est euclidienne.

• Une relation R est symétrique, transitive et reproductive ssi
elle est réflexive et euclidienne, ou encore, ssi
elle est réflexive, symétrique et transitive.
(Ce sont donc trois façons de caractériser les relations d'équivalence.)