Représentation des connaissances et logique modale : Cours

Master 2 MICR

Institut Galilée

François Lévy

5 mars 2006

Plan

1. Cours 1 (26 sept) Logique propositionnelle : langage, connecteurs et raisonnement, valeur de vérité, modèle ; tautologie, satisfiabilité
2. Cours 2 (3 oct) Logique propositionnelle : preuve, axiomatisations, notion de démonstration automatique. Résolution, clause de Horn
3. Cours 3 (10 oct) Logique des prédicats : langage et pouvoir d'expression, formules closes et variables libres, modèle, validité dans un modèle, tautologies.
4. Cours 4 (17 oct) Logique des prédicats : axiomes, preuves. Quelques remarques sur la démonstration automatique (unification, faux premier ordre)
5. Cours 5 (24 oct) Retour sur le pouvoir de représentation : assertion, interrogation, nécessité et contingence, obligation, savoir. Formalisme modal propositionnel. Discussion de la vraisemblance de quelques formules particulières.
6. Cours 6 (31 oct) Sémantique des logiques modales propositionnelles. Modèle et cadre. Les logiques de base.
7. Cours 7 (7 nov) Axiomatisations et preuves en logiques modales propositionnelles. Dualité.
8. Cours 8 (14 nov) Propriété de réduction des chaînes de modalités. Démonstration automatique : méthode des tableaux.
9. Cours 9 (21 nov) Un cas particulier intéressant : les logiques temporelles. Structures du temps et propriétés logiques.

Cours 1

Logique propositionnelle : le point de vue sémantique

1.1 Intuitions

1.1.1 propositions

Qu'est-ce qu'une proposition ? Quelque chose qui est **vrai** ou **faux** !

Ex : "il pleut", "il fait beau", "l'herbe est mouillée", "après le repas, je tonds la pelouse", "Il y a un bon film à la télévision ce soir", ...

Quand on a une liste de propositions, on peut fabriquer d''autres énoncés, par ex. "il fait beau **ou** l'herbe est mouillée". Ce nouvel énoncé peut être vrai ou faux (donc c'est aussi une proposition, au sens précédent). C'est la mécanique de base du calcul propositionnel : fabriquer de nouvelles propositions à partir de celles qui existent.

La fabrication de nouvelles propositions : Pr ex : "Il pleut **et** il y a un bon film à la télévision ce soir", "Je **ne** tonds **pas** la pelouse après le repas", "**s'**il pleut cet après-midi**,** l'herbe sera mouillée", "l'herbe sera mouillée **seulement s**'il pleut cet après-midi", "il pleut **si et seulement si** il **ne** fait **pas** beau"

Les symboles du calcul propositionnel On *nomme* chaque proposition élémentaire, ex :
- p1 = "il pleut",
- p2 ="il fait beau",
- p3 = "l'herbe est mouillée",
- p4 = "après le repas, je tonds la pelouse",
- p5 = "Il y a un bon film à la télévision ce soir"

On a une *liste finie* de **connecteurs** qui permettent de former de nouveaux énoncés. A chacun est attaché son *arrité* (le nombre d'arguments dont il a besoin) :

$\neg$ (non : 1) $\neg$p4 = "Je **ne** tonds **pas** la pelouse après le repas"
$\wedge$ (et : 2) p1 $\wedge$ p5 = "Il pleut **et** il y a un bon film à la télévision ce soir"
$\vee$ (ou : 2) p1 $\vee$ p2
$\rightarrow$ (implique : 2) p6 $\rightarrow$ p7 = "s'il pleut cet après-midi, l'herbe sera mouillée"
$\leftrightarrow$ (équivaut :2) p1 $\leftrightarrow \neg$p2 = "il pleut si et seulement si il ne fait pas beau"

Remarques Une proposition élémentaire s'appelle un **atome**. Il n'y a pas d'autre façon de créer des propositions que de les construire avec les connecteurs. Ce qui veut dire qu'en logique des propositions, "il pleut" et "il pleut cet après-midi" sont deux propositions totalement indépendantes (p1 et p6), de même que "l'herbe est mouillée" et "l'herbe sera mouillée" (p3 et p7).

Deux propositions atomiques sont soit identiques (deux noms pour la même proposition), soit sans rapport.

Une proposition qui est soit un atome, soit la négation d'un atome s'appelle un **litéral**. On a utilisé $\neg$p4 dans les exemples. On remarquera qu'avant de donner cette formalisation, il a fallu faire un travail de traduction : "Je ne tonds pas la pelouse après le repas" n'est pas "non après le repas, je tonds la pelouse"!! C'est une question que nous retrouverons souvent : *dans quelle mesure la traduction en logique que nous faisons est-elle conforme à l'original ?*

Dernière remarque, tous les asemblages de connecteurs, d'atomes et de parenthèses (ces dernières utilisées pour désambiguiser, puisqu'on utilise une notation infixée) ne sont pas des propositions : si une formule n'est pas *bien formée*, se demander si elle représente un énoncé vrai ou faux n'a pas de sens. Par exemple, (p1 $\neg\wedge$ p2) ou ($\wedge$ p1 $\vee$ p2) ne sont pas bien formées parce que les arités ne sont pas respectées, (p1 $\wedge$ p2 $\vee$ p3) n'est pas bien formée parce qu'il y a deux façons de la lire.

1.1.2 Propositions bien formées

Si l'on veut qu'un ordinateur puisse faire du calcul des propositions, la première tâche est de reconnaître les propositions qui ont un sens. On utilise pour cela une définition des formules bien formées.

Définition 1 *L'ensemble des propositions bien formées (pbf) est le plus petit ensemble tel que :*

- un atome est une pbf
- si P, P1 et P2 sont des pbf, alors
  - $\neg$P est une pbf
  - (P1 $\wedge$ P2) est une pbf
  - (P1 $\vee$ P2) est une pbf
  - (P1 $\rightarrow$ P2) est une pbf
  - (P1 $\leftrightarrow$ P2) est une pbf

Remarquez que cette définition fournit implicitement un algorithme pour décider si une formule est bien formée. Remarquez aussi qu'elle est très restrictive sur l'usage des parenthèses : on doit écrire (p1 $\wedge$ (p2 $\wedge$ p3)) ou ((p1 $\wedge$ p2) $\wedge$ p3). Pour lever cette contrainte, il faudra d'abord montrer que le $\wedge$ est associatif ...

1.1.3 Le rôle des connecteurs

On en vient alors à la question qui nous motivait au départ : quand peut-on dire qu'une pbf est vraie ou fausse ? Pour cela, on répond en détail pour chacune des règles de formation de pbf plus complexe.

Les règles du non, du et et du ou sont assez intuitives. Celle de l'implication l'est moins. En gros la question est : quand on dit "s'il vient, je mange mon chapeau", on comprends générallement que s'il ne vient pas, je ne mangerai pas ... Pour "s'il pleut, je mets mon imperméable", c'est un peu moins flagrant, mais quand même. Donc, dans la pratique quotidienne, *si* n'est pas distingué de *si et seulement si*. Pour le logicien, on n'a rien dit de ce qui se passe "s'il ne vient pas" ou "s'il ne pleut pas" – donc s'il ne pleut pas et que je mets mon imperméable, je n'avais pas anoncé quelque chose de faux ; et donc la proposition ne peut être que vraie dans ce cas. D'où les règles :

- $\neg$P est vrai si P est faux ; sinon, $\neg$P est faux.
- (P1 $\wedge$ P2) est vrai si P1 est vrai **et** P2 est vrai ; sinon, (P1 $\wedge$ P2) est faux.
- (P1 $\vee$ P2) est vrai si **au moins un** des P1, P2 est vrai ; (P1 $\vee$ P2) est faux sinon
- (P1 $\rightarrow$ P2) est vrai **soit si** P1 est faux **soit si** P2 est vrai ; sinon, (P1 $\rightarrow$ P2) est faux.
- (P1 $\leftrightarrow$ P2) est vrai si P1 et P2 sont **tous les deux** vrais ou **tous les deux** faux ; sinon, (P1 $\leftrightarrow$ P2) est faux.

Remarquez bien que, quand on formule ces règles, elles s'appliquent à *n'importe quelle* proposition bien formée P, P1, P2. On a donc introduit en passant des **variables propositionnelles**. Ces règles peuvent se résumer dans des tableaux, que l'on appelle *tables de vérité*. Par ex :

$\neg$	V	F
	F	V

P $\neg$P1

$\wedge$	V	F
V	V	F
F	F	F

P1 $\wedge$ P2

$\vee$	V	F
V	V	V
F	V	F

P1 $\vee$ P2 .

$\rightarrow$	V	F
V	V	F
F	V	V

P1 $\rightarrow$ P2

$\leftrightarrow$	V	F
V	V	F
F	F	V

P1 $\leftrightarrow$ P2 .

1.2 Interprétation, satisfiabilité, tautologies

On ne sait toujours pas si une proposition composée est vraie ou fausse. A vrai dire, la question est mal posée : ça dépend des propositions atomiques ... Nous allons voir cela plus en détail.

On peut voir chaque connecteur comme une fonction à 1 ou deux arguments, et une pbf comme une fonction composée. On vérifie à l'aide des tables de vérité que $\wedge$ et $\vee$ sont chacun commutatif et associatif. Quand deux pbf sont identiques en tant que fonction, on peut omettre les parenthèses. Techniquement, on n'a pas besoin de changer la définition des fbf – on laisse seulement au lecteur le choix entre deux pbf qui représentent la même fonction.

Le choix d'un ensemble de valeurs pour les propositions atomiques – "il pleut" est vraie, "l'herbe est mouillée" aussi, "il y a un bon film à la télévision ce soir " est faux." – s'appelle une **interprétation**. Du point de vue sémantique, c'est toujours par rapport à une interprétation qu'on peut dire si une proposition est vraie ou fausse. Il faut pour bien comprendre faire la différence entre la proposition atomique et sa valeur dans l'interprétation : la proposition est p, p1, p2, ... ; sa valeur peut-être V ou F, et donc V, F NE SONT PAS des propositions. Par contre, on a besoin de deux propositions spéciales dont la valeur est la même dans toutes les interprétations. On appelle ces propositions $\top$et $\bot$.

Définition 2 *Une interprétation $\mathcal{I}$ est une fonction de l'ensemble des propositions atomiques dans $\{V, F\}$, telle que $\mathcal{I}(\top)=$ V et $\mathcal{I}(\bot)=$ F.*

On dit que l'interprétation $\mathcal{I}$ *satisfait* la pbf P si la valeur de P (vue comme une fonction) appliquée à $\mathcal{I}$ est V. On écrit $\mathcal{I} \models$P. La définition exacte est calquée sur celle des pbf, puisqu'il s'agit d'étendre $\mathcal{I}$ à toutes les pbf.

Définition 3 *$\mathcal{I}$ satisfait P, ou encore P est valide dans $\mathcal{I}$ (noté $\mathcal{I} \models P$) si et seulement si l'une des conditions suivante est vérifiée :*

- P est un atome et $\mathcal{I}(P) =$ V
- P = $\neg$P' et $\mathcal{I} \not\models$ P'
- P = P1 $\vee$ P2 et soit $\mathcal{I} \models$ P1, soit $\mathcal{I} \models$ P2
- P = P1 $\wedge$ P2 et à la fois $\mathcal{I} \models$ P1 et $\mathcal{I} \models$ P2
- P = P1 $\rightarrow$ P2 et soit $\not{\mathcal{I}} \models P1$, soit $\mathcal{I} \models$ P2
- P = P1 $\leftrightarrow$ P2 et soit $\mathcal{I} \models$ P1 et $\mathcal{I} \models$ P2, soit $\mathcal{I} \models \neg$P1 et $\mathcal{I} \models \neg$P2

Attention, il faut bien distinger les valeurs V, F (représentant le vrai et le faux **en tant qu'objets du calcul proposotionnel** et les termes "vrai", "faux" **naïfs** qu'on emploie par exemple pour expliquer un résultat en disant "$\mathcal{I}$ satifait P est [ou : a l'air] faux" - on est dans le métalangage pour parler du calcul.

On s'intéresse aussi à la question inverse de la précédente : étant donnée une pbf, quelles sont les intepprétations qui la satisfont ? On a alors deux cas particuliers intéressants :

Définition 4 *la pbf P est* satisfiable *ssi il existe au moins une interprétation $\mathcal{I}$ telle que $\mathcal{I} \models P$. Elle est une* tautologie *si toutes les interprétations vérifient $\mathcal{I} \models P$.*

Les deux notions sont étroitement liées. Comme toute interprétation satisfait soit P, soit ¬P, P est satisfiable si ¬P n'est pas une tautologie, et P est une tautologie si ¬P n'est pas satifiable.

1.3 Modèle, conséquence sémantique

1.4 Variantes

Nous avons donné ici une présentation des connecteurs, propositions bien formées, interprétations. Ce n'est pas la seule présentation possible. Nous allons rapidement en indiquer une autre. Les deux sont équivalentes, au sens où elles conduisent aux mêmes formules bien formées et aux mêmes tautologies (vous pourrez le démontrer en exercice). Il est important de comprendre quelle est la différence, parce que cela implique une réflexion sur les écritures employées qui devrait vous aider à faire des démonstrations. De plus, suivant le travail à faire, il peut être plus intéressant d'adopter l'une ou l'autre, et c'est utile de pouvoir simplifier les preuves.

– Atomes propositionnels : les mêmes
– Connecteurs : → (arité 2), **f** (arité 0)

Des exemples de fbf sont : (p1 → p2), ((p1 → p2) → p3), (p1 → (p2 → p3)), mais aussi **f**, (p1 → **f**), ((p1 → **f**) → p3), etc.

Qu'en est-il des autres connecteurs ? ce sont des *abréviations*. Autrement dit, on a le droit de se servir de ces symboles pour raccourcir la notation et rendre plus lisibles des formules complexes. Mais ils n'ont pas de sens propre : une proposition complexe avec des ∧ , ∨ , etc. est bien formée (resp. valide) si et seulement si la formule qu'elle abrège (avec seulement → et **f**) est bien formée (resp valide). Voici les abréviations :

– **v** est une abbréviation pour (**f** → **f**)
– ¬P est une abbréviation pour (P → **f**)
– (P1 ∨ P2) est une abbréviation pour ((P1 → **f**)→ P2)
– (P1 ∧ P2) est une abbréviation pour ((P1 → (P2 → **f**))→ **f**).
– (P1 ↔ P2) est une abbréviation pour (((P1 → P2) → ((P2 → P1) → **f**))→ **f**)

La définition des pbf et de l'interprétation devient :

– **f** est une pbf
– un atome est une pbf
– si P1 et P2 sont des pbf, alors
  – (P1 → P2) est une pbf
– rien d'autre n'est une pbf

Définition 5 *$\mathcal{I}$ satisfait P (noté $\mathcal{I} \models P$) si et seulement si l'une des conditions suivante est vérifiée :*

– P est un atome et $\mathcal{I}(P) = \mathrm{V}$
– P = P1 → P2 et soit $\not\mathcal{I} \models P1$, soit $\mathcal{I} \models$ P2

Notez qu'il s'ensuit de cette définition qu'aucune interprétation ne satisfait **f**.

1.5 Démonstration automatique

Soit une propositions A. On va chercher s'il y a une interprétation qui satisfait ¬A, en appliquant systématiquement les règles de 3 à ¬A.

On construit un tableau sémantique, i.e. un arbre dont les noeuds comportent une ou plusieurs formules. Initialement, il contient seulement une racine avec une seule formule. La formule à la racine est la négation de A (la formule à prouver). L'arbre est étendu progressivement par application des règles ci-dessous. Dans certains cas, une règle crée plusieurs branches, mais dans le cas les plus simples elle n'en introduit qu'une.

Chaque feuille contient une ensemble de propositions tel que toute interprétation qui satisfait cet ensemble satisfait la racine. L'objectif est de fermer chaque branche par une contradiction

- c'est à dire, parce que deux noeuds sur la feuille (dans la branche si l'on implémente plus économiquement) contiennent l'un une formule et l'autre sa négation. Si toutes les branches sont fermées, $\neg$A n'est pas satisfiable et donc A est prouvée. Les règles s'appliquent uniquement sur les feuilles et sont les suivantes ("ajouter un (deux) noeud" veut toujours dire l'ajouter (les ajouter) en tant que fils du noeud sur lequel on travaille) :

- Si le noeud contient $\neg\neg$P ajouter un noeud remplaçant cette formule par P
- Si le noeud contient (P $\wedge$ Q) ajouter un noeud en remplaçant cette formule par P, Q
- Si le noeud contient $\neg$(P $\vee$ Q) ajouter un noeud remplaçant cette formule par $\neg$P, $\neg$Q
- Si le noeud contient $\neg$(P $\rightarrow$ Q), ajouter un noeud en remplaçant cette formule par P, $\neg$Q.
- Si le noeud contient $\neg$(P $\wedge$ Q) ajouter deux noeuds, un remplaçant cette formule par $\neg$P et l'autre par $\neg$Q
- Si le noeud contient (P $\vee$ Q) ajouter deux noeuds, un remplaçant cette formule par P et l'autre par Q
- Si le noeud contient (P $\rightarrow$ Q) ajouter deux noeuds, un remplaçant cette formule par $\neg$P et l'autre par Q (on pourrait enlever cette règle, puisque P $\rightarrow$ Q est ($\neg$P) $\vee$ Q ; elle fait gagner du temps.
- Si le noeud contient (P $\leftrightarrow$ Q) ajouter deux noeuds. Un remplace cette formule par P, Q et l'autre par $\neg$P, $\neg$Q
- Si le noeud contient $\neg$(P $\leftrightarrow$ Q) ajouter deux noeuds. Un remplace cette formule par P,$\neg$Q et l'autre par $\neg$P, Q

Cours 2

Logique propositionnelle : point de vue syntaxique

2.1 Intuitions

Nous avons vu que certaines propositions sont universellement valides, ou conséquences les unes des autres, en considérant les interprétations qui les satisfont. C'est même grâce à cette notion d'interprétation que nous avons construit une définition rigoureuse de ces termes : c'est ce qu'on appelle le point de vue sémantique. Une autre approche est possible qui ne se sert pas du tout de la notion d'interprétation. Au lieu de définir la valeur d'une proposition, on considère les écritures (propositions atomiques, connecteurs) comme des *objets* que l'on peut manipuler avec des règles, sur lesquels on peut calculer comme sur des variables dans une équation. C'est déja ce que nous avons fait pour définir les pbf ; dans ce chapitre, on ne garde du précédent que la définition 1, et on étudie des règles qui, supposant que certaines pbf sont démontrées, permettent d'en démonter d'autres. Ces règles s'appellent des règles d'inférence. Comme il faut bien un point de départ, on a aussi un certain nombre de pbf qui sont données sans démonstration : les axiomes.

Il est important de bien comprendre qu'on construit une autre théorie : d'une part techniquement, le raisonnement n'est pas compréhensible sans prendre conscience de cet 'oubli' du chapitre 1, non plus que la question qui se posera de comparer les deux approches. D'autre part, quand on cherchera à compliquer le calcul logique pour rendre compte de plus de choses, l'analyse des insuffisances et les idées pour y remédier ne sont pas les mêmes dans les deux points de vue.

2.2 Axiomes et règles d'inférence

Un dernier point à préciser : au sens strict, un axiome est une pbf, donc ne contient pas d'autre symboles que des atomes et des connecteurs. N'importe quelle théorie a une infinité d'axiomes tous semblables le uns aux autres (vous allez comprendre pourquoi de suite). Pour éviter une infinité de formules, les lettres P, Q, R ... ci-dessous sont des *variables de pbf* : on peut leur substituer n'importe quelle pbf. Une formule (règle) employant des variables de pbf s'appelle un **schéma d'axiome (de règle)**. Par exemple, $\mathrm{P} \to (\mathrm{Q} \to \mathrm{P})$ vaut pour $(p0 \to\ (p1 \to\ p0))$ ou $((p0 \wedge\ p1) \to\ ((p1 \vee\ p2) \to\ (p0 \wedge\ p1))$, etc.

les axiomes

Propriétés de l'implication :

$\mathrm{P} \to (\mathrm{Q} \to \mathrm{P})$ **(PROP1)**

$(\mathrm{P} \to (\mathrm{Q} \to \mathrm{R})) \to ((\mathrm{P} \to \mathrm{Q}) \to (\mathrm{P} \to \mathrm{R}))$ **(PROP2)**

Propriétés de la conjonction

$\mathrm{P} \wedge \mathrm{Q} \to \mathrm{P}$ **(PROP3)**

$P \wedge Q \rightarrow Q$ **(PROP4)**
$P \rightarrow (Q \rightarrow P \wedge Q)$ **(PROP5)**
Propriétés de la disjonction
$P \rightarrow P \vee Q$ **(PROP6)**
$Q \rightarrow P \vee Q$ **(PROP7)**
$(P \rightarrow R) \rightarrow ((Q \rightarrow R) \rightarrow (P \vee Q \rightarrow R))$ **(PROP8)**
Propriétés de la négation
refutation (Reductio ad absurdum) :
$(P \rightarrow Q) \rightarrow ((P \rightarrow \neg Q) \rightarrow \neg P)$ **(PROP9)**
non contradiction (Ex falso aliquid sequitur) :
$\neg P \rightarrow (P \rightarrow Q)$ **(PROP10)**
tiers exclu (Tertium non datur) :
$P \vee \neg P$ **(PROP11)**

une règle d'inférence Il n'y en a qu'une, mais elle est essentielle : c'est la seule façon de démontrer de nouvelles propositions. Nous introduisons pour cela le symbole $\vdash$: $(p1 \vee p2), (\neg p2 \vee p3) \vdash p1 \vee p3$ signifie " $(p1 \vee p2)$ et $(\neg p2 \vee p3)$ suffisent pour une démonstration de $p1 \vee p3$ (ce que nous démontreront être le cas plus tard).

Le schéma de règle d'inférence du calcul propositionnel est donc :

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

(MP pour Modus Ponens) dans lequel P et Q sont des variables propositionnelles. Les pbf à gauche et à droite d'une instanciation de (MP) sont ses prémisses et sa conclusion. On peut alors définir simplement les théorèmes :

Définition 6 *L'ensemble des théorèmes est le plus petit ensemble vérifiant :*
- *un axiome est un théorème*
- *si les prémisses d'une instanciation de (MP) sont des théorèmes, sa conclusion est un théorème.*

2.3 Démonstration

On peut construire des démonstrations basées sur l'approche syntaxique. Les méthodes de démonstration ci-dessous ont un intérêt mathématique parce qu'elles permettent de vérifier la correction d'une démonstration ; ce sont aussi les premières ébauches des techniques de démonstration automatique, et s'il y a eu des améliorations techniques de ce point de vue, l'essentiel des problèmes est déja présent.

2.3.1 La déduction naturelle (Hilbert)

Une démonstration selon la méthode de Hilbert de la proposition Q est une suite finie de propositions P0, P1, . . . , Pm telle que Pm = Q et pour tout i $\leq$ m
- soit Pi est un axiome,
- soit Pi est la dernière proposition d'une autre démonstration,
- soit il existe j, k < i tels que MP s'applique à Pj, Pk et Pi est la conclusion de cette instanciation de MP.

Remarquez que le second item est en théorie supperflu : il suffirait de mettre les deux démonstrations bout à bout. En pratique, il simplifie bien la présentation des démonstrations. Pour qu'elles soient lisibles, on place une proposition par ligne et on marque chaque proposition par la règle qui la justifie.

On peut montrer que Q est un théorème si et seulement si il existe au moins une démonstration de Q selon la méthode de Hilbert. Le *si* se démontre sans difficulté (montrer par récurence sur son rang que toute proposition de la démonstration est un théorème). Pour la réciproque, il faut montrer que tout théorème est la conclusion d'une démonstration (indication : considérez l'ensemble des conclusions de toutes les démonstrations possibles).

On peut simplifier les démonstrations en utilisant l'introduction d'hypothèse : on introduit (sans condition) dans la démonstration une proposition H quelconque marquée *hypothèse*, et on l'utilise comme les autres formules. Par la suite, après une proposition P, on *relève* l'hypothèse H en introduisant la formule H → P. Appelons *démonstration étendue* une démonstration dans laquelle on a la possibilité d'introduire et de relever des hypothèses.

Une démonstration étendue est correcte si toutes les hypothèses ont été relevées, et si toute hypothèse H1 introduite à l'intérieur de H est relevée avant H. On montre que, si elles simplifient la démonstration, les démonstrations étendues ne sont pas fondamentalement différentes des autres :

Théorème 1 *Si une proposition a une démonstraion étendue correcte, elle a une démonstration standard.*

Pour *prouver* ce théorème, on donne un algorithme qui transforme la démonstration étendue en démonstration standard (noter la différence entre preuve et démonstration). Pour chaque ligne (numérotée n) contenant la proposition P et écrite sous l'hypothèse H dans la démonstration étendue démonstration étendue, on trouve une ligne (numérotée $corr\, n$) contenant H → P dans la démonstration standard. Pour celà, on applique à chaque ligne de la démonstration étendue entre l'introduction de H et la ligne où H est relevée, dans l'ordre de lecture, les règles suivantes :

- si la ligne de démonstration étendue contient H, écrire l'axiome H → H
- si la ligne n de démonstration étendue est un axiome ax, écrire :
  [$corr\, n - 2$] Ax
  [$corr\, n - 1$] (Ax → (H → Ax)) [PROP1]
  [$corr\, n$] (H → Ax) [n, $n + 1$, MP]
- Si la ligne de démonstration étendue et ses références sont de la forme :
  [$n1$] P
  [$n2$] P → Q
  [n] Q [$n1$, $n2$, MP]
  Dans ce cas, si $n1$ ou $n2$ sont avant l'introduction de H dans la démonstration, on applique à ces lignes la même méthode que pour les axiomes (en omettant $corr\, n - 2$ qui est redondant). Maintenant, quels que soient $n1$ et $n2$, par hypothèse de récurence, quand on traite la ligne n, on a déja les lignes [$corr\, n1$] H → P et [$corr\, n2$] H → (P → Q). On ajoute les lignes :
  [$corr\, n - 2$] (H → (P → Q)) → ((H → P) → (H → Q)) [PROP2]
  [$corr\, n - 1$] (H → P) → (H → Q) [$corr\, n - 2$, $corr\, n2$, MP]
  [$corr\, n$] H → Q [$n1$, $corr\, n - 1$, MP]

Il est clair qu'à chaque ligne n l'algorithme s'applique et produit bien une ligne $corr\, n$ qui a la propriété requise. Maintenant, en appliquant répétitivement l'algorithme à toutes les hypothèses depuis les plus intérieures vers les extérieures, on obtient une démonstration standard.

2.3.2 Les séquents (Genzen)

On peut voir la notation de Gentzen comme un commentaire sur celle de Hilbert : on écrit H1, H2, ... Hk ⊢P1, P2 ... Pm pour indiquer que, en utilisant les hypothèsees Hi, on peut démontrer la propositions P = (P1 ∨ ... ∨ Pm). Ensuite, on remarque que si on sait qu'il existe une telle démonstration, certaines transformations simples permettent de déduire d'autres démonstrations. Traditionellement, Γ et les majuscules greques désignent des ensembles de propositions. On a par exemple :

$$\frac{\Gamma \vdash A, \Gamma \vdash B}{\Gamma \vdash A \wedge\ B}$$

c'est à dire : s'il existe une démonstration de A et une autre de B utilisant toutes deux les hypothèses Γ, alors il existe une démonstration de $A \wedge\ B$ utilisant Γ.

La méthode des séquents propose un ensemble de transformations permettant de générer toutes les descriptions de démonstrations possibles seulement à partir démonstrations triviales de la forme A ⊢A. Les transformations (ou règles) de Gentzen sont les suivantes :

Règles structurelles

affaiblissement	contraction	échange	cut
$\dfrac{\Gamma \vdash \Lambda}{A, \Gamma \vdash \Lambda}$ $\quad$ $\dfrac{\Gamma \vdash}{\Gamma \vdash A}$	$\dfrac{A, A, \Gamma \vdash \Lambda}{A, \Gamma \vdash \Lambda}$	$\dfrac{\Pi, A, B, \Gamma \vdash \Lambda}{\Pi, B, A, \Gamma \vdash \Lambda}$	$\dfrac{\Gamma \vdash C; C, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Lambda}$

Règles logiques

	gauche	droit
$\wedge$	$\dfrac{A, \Gamma \vdash \Lambda}{A \wedge B, \Gamma \vdash \Lambda}$ $\quad$ $\dfrac{B, \Gamma \vdash \Lambda}{A \wedge B, \Gamma \vdash \Lambda}$	$\dfrac{\Gamma \vdash A, \Gamma \vdash B}{\Gamma \vdash A \wedge B}$
$\vee$	$\dfrac{A, \Gamma \vdash \Lambda; B, \Gamma \vdash \Lambda}{A \vee B, \Gamma \vdash \Lambda}$	$\dfrac{\Gamma \vdash A}{\Gamma \vdash A \vee B}$ $\quad$ $\dfrac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$
$\neg$	$\dfrac{\Gamma \vdash A}{\neg A, \Gamma \vdash}$	$\dfrac{A, \Gamma \vdash}{\Gamma \vdash \neg A}$
$\rightarrow$	$\dfrac{\Gamma \vdash A; B, \Pi \vdash \Lambda}{A \rightarrow B, \Gamma, \Pi \vdash \Lambda}$	$\dfrac{A, \Gamma \vdash B}{\Gamma \vdash A \rightarrow B}$

2.4 Alternatives

Ici aussi, on peut donner bien d'autres présentations. Nous continuons la même alternative que dans la version sémantique (celle qui considère tous les connecteurs sauf → comme des abréviations) en en donnant une axiomatique qui a été proposée par Hilbert :

P → (Q → P) **(HPROP1)**
(P → (Q → R)) → ((P → Q) → (P → R)) **(HPROP2)**
(¬Q → ¬P) → (P → Q) **(HPROP3)**

2.5 Démonstration automatique

La première méthode de démonstration automatique syntaxique, due à Robinson, est basée sur le **principe de résolution**. Elle est encore très utilisée, et a donné lieu à de nombreuses variantes. Dans le formalisme de Gentzen, le principe de résolution s'énonce :

$$\frac{\Gamma \vdash A, \Lambda; \Gamma \vdash \neg A, \Pi}{\Gamma \vdash \Lambda, \Pi}$$

(rappelez-vous qu'à droite du signe ⊢on a implicitement une disjonction).

2.5.1 Forme normale, clause

La mise en place du principe de résolution dépend de la *normalisation* des formules : pour pouvoir comparer plus facilement si deux formules sont identiques ou pas, on choisit un standard d'écriture (on dit une *forme normale*) qui facilite la comparaison et le calcul. On écrit les formules avec uniquement ¬, ∧ , ∨ (les autres connecteurs sont traités comme des abréviations et expansés). Ensuite, on utilise systématiquement les trois règles suivantes :

$$\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$$

$$\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$$

$$A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$$

La formule est mise en forme normale conjonctive : les négations sont ammenées le plus à l'intérieur possible, jusqu'au niveau des propositions ; les connecteurs les plus à l'extérieur (les plus près de la racine si on voit une formule comme un arbre) sont les conjonctions, qui gouvernent les disjonctions, qui gouvernent les négations. De façon plus formelle, une formule en forme normale conjonctive s'écrit :

$$\bigwedge_{i=1}^{k} (\bigvee_{j=1}^{m} l_{i,j})$$

où $l_{i,j}$ est un litéral, c'est à dire un atome propositionnel ou sa négation.

Une proposition qui est une disjonction de litéraux $(\bigvee_{j=1}^{m} l_{i,j})$ s'appelle une **clause**. Une formule mise ne forme clausale peut être traitée comme un ensemble de clauses, et comme une théorie est un ensemble de formules, c'est aussi un ensemble de clauses.

2.5.2 Résolution

Dans les méthodes de démonstration automatique basées sur la résolution, on a un ensemble d'hypothèses (une théorie T) et on veut démontrer une proposition donnée. D'abord on met toutes les formules de la théorie en forme clausale. Ensuite, on compare systématiquement les clauses deux à deux, à la recherche de litéraux complémentaires, et l'on applique la règle suivante (pour simplifier l'écriture, on suppose ici les litéraux complémentaires en tête, mais l'ordre est en fait quelconque) : si deux clauses de T sont de la forme $(l \bigvee_i l_i)$ et $(\neg l \bigvee_j l_j)$, ajouter à T la clause $\bigvee_i l_i \vee \bigvee_j l_j$.

Appliquée à l'aveugle, la méthode est coûteuse en temps de calcul. Les perfectionnements relèvent à la fois d'heuristiques plus ou moins complètes et d'implémentations efficaces pour la recherche des litéraux complémentaires et le parcours des ensembles de clauses.

Cours 3

Logique des prédicats 1

3.1 Intuitions

On revient maintenant sur une lacune de la logique des propositions que nous avions constatée au chapitre 1 : “Sa voiture est rouge” et “sa voiture est jaune” sont deux propositions totalement indépendantes., de même que “Jean prend son manteau” et “Pierre prend son manteau”. La logique des prédicats est destinée à prendre en charge cette structure interne des propositions.

L'idée remonte aux grecs anciens, et se formule ainsi : dans une phrase, le verbe établit une relation entre son sujet et ses compléments. La formulation mathématique fait quand même l'impasse sur la différence de rôle entre le sujet et les différents compléments, pour ne garder que le rôle central de la relation : on écrira $Prend\,(\mathrm{Jean}, \mathrm{manteau})$ - ce qui laisse quand même une trace qu'il y a quelque chose de commun avec $Prend\,(\mathrm{Pierre}, \mathrm{Manteau})$ ou $Prend\,(\mathrm{Jean}, \mathrm{chapeau})$. Le premier exemple est souvent traduit par $Couleur\,(\mathrm{voiture}, \mathrm{rouge})$, parce que le verbe être ne suffit pas à caractériser une relation – pour des raisons plus philosophiques ou linguistiques que mathématiques, le problème se reposera peut-être à vous plus tard.

On a donc fait un progrès dans le pouvoir d'expression par rappport aux chapitres précédents : on peut maintenant raisonner sur les arguments de la relation. Premier point, on va avoir un ensemble d'objets pour jouer ce rôle d'argument, et cet ensemble d'objets pourra avoir une structure assez riche. Pour rester au minimum, il sera muni de fonctions – on pourra par exemple distinguer les manteaux de Pierre et de Jean en utilisant une fonction manteau() dont l'argument est une personne.

Second élément que nous n'introduirons pas ici, mais qui est facile à ajouter : on peut typer les prédicats et les objets, c'est à dire distinguer différentes catégories d'objets et contraindre la catégorie de chaque argument des relations. De cette façon, $Couleur\,(\mathrm{voiture}, \mathrm{rouge})$ est acceptable et $Couleur\,(\mathrm{rouge}, \mathrm{voiture})$ ne l'est pas.

Dernier point qui va jouer un rôle essentiel, on n'a un gain réel dans le pouvoir d'expression que parce qu'on introduit deux quantificateurs : l'universel ($\forall$) et l'existentiel ($\exists$). Ils permettent d'écrire des propriétés valides pour tous les objets et donc, avec de minuscules astuces techniques, spécifiques à tel ou tel ensemble d'objets que l'on sait caractériser par une propriété - ceux que l'on appelle les ensembles définissables. Autrement dit, on est capable de raisonner sur des objets qui n'ont pas de nom.

Une question de vocabulaire enfin. Il peut sembler arbitraire de limiter les quantificateurs à porter sur les objets. Quantifier sur les relations a du sens (quelle que soit la Relation Ami() de ma logique, elle s'applique au moins à Castor et Pollux). Une telle logique s'appelle une logique des propositions du 2eme ordre. Les démonstrations y sont très difficiles et il y a peu de résultats. Quand nous disons “logique des prédicats”, nous sous-entendons en général : du 1er ordre (en abrégé LPO). La logique des propositions peut être assimilée à une logique d'ordre 0.

3.2 Langage

3.2.1 Symboles

Le langage de la LPO comporte :
- un ensemble dénombrable de symboles de fonction. Chaque symbole de fonction a une arité.Les symboles de fonction d'arité 0 sont appelés symboles de constante. On note les fonctions par c, f, g, ...
- un ensemble dénombrable de symboles de variables, notés x, y, ...
- un ensemble dénombrable de symboles de relation. Chaque symbole de relation a une arité. On utilise les lettres P, Q, R, ...
- un ensemble fini de connecteurs : ceux de la logique des propositions, auxquels on ajoute deux connecteurs à deux places, les quantificateurs $\forall$et $\exists$.

3.2.2 Termes

Les termes vont servir à désigner les objets sur lesquels portent les relations. On les définit récursivement : l'ensemble des termes est le plus petit ensemble tel que
- toute variable est un terme
- si f est une fonction d'arité n et $t_1, \ldots t_n$ sont des termes, alors $f(t_1, \ldots t_n)$ est un terme.

Notez que le second item implique que toute constante est un terme. Remarquez de suite que les termes sont des *noms* d'objets, et que leur construction ne s'engage pas sur le fait que deux noms d'objets différents désignent le même objet ou deux objets différents.

3.2.3 Atomes, litéraux, formules

Les atomes jouent dans la construction à peu près le même rôle que les propositions atomiques en logique de propositions (à peu près à cause des quantificateurs). C'est toujours le plus petit élément auquel on puisse assigner la valeur vrai ou faux.

Définition : un atome ou formule atomique est une expression de la forme $R(t_1, \ldots, t_n)$ où R est un symbole de relation d'arité n et $t_1, \ldots, t_n$ sont des termes. Un litéral est un atome ou la négation d'un atome.

Nous en arrivons maintenant à la définition des formules bien formées. La nouveauté par rapport à la LP est l'usage de quantificateurs. Leur premier argument est un symbole de variable, le second est une formule bien formée (définie ci-dessous, on vérifiera qu'il n'y a pas de cercle vicieux dans la définition). Quand on écrit $\forall x F(x)$ ou $\forall x(F(x) \wedge G(x))$, x est le premier argument du quantificateur, F(x) ou $(F(x) \wedge G(x))$ son second argument (les parenthèses sont indispensables dans le second cas pour désambiguiser).

Définition 7 *L'ensemble des formules bien formées (fbf) est le plus petit ensemble tel que :*
- *tout atome est une fbf*
- *si F, F1 et F2 sont des fbf,* $\neg F$, $F1 \wedge F2$, $F1 \vee F2$, $F1 \rightarrow F2$, $F1 \leftrightarrow F2$ *sont des fbf*
- *si F est une fbf,* $\forall xF$ *et* $\exists xF$ *sont des fbf*

3.2.4 variable liée, libre

La *portée* d'un quantificateur est son second argument. Une occurence de la variable x dans la formule Φ est **liée** dans Φ si elle est dans la portée d'un quantificateur de Φ dont le premier argument est x. Autrement, elle est libre dans Φ.

On verra que la notion de variable libre / liée dans une formule ou une sous-formule est importante pour la manipulation correcte des quantificateurs. On dit aussi d'une variable x liée dans Φ qu'elle est *muette* : intuitivement, la vérité de Φ ne dépend pas de la valeur de x.

Une **formule close** est une fbf qui n'a pas de variable libre (dont toutes les variables sont liées).

3.3 Univers, interprétation, modèles

On passe maintenant à la définition de la sémantique des fbf. Comme on a donné au langage plus de pouvoir d'expression, on ne va plus être limité à exprimer le vrai ou le faux : les prédicats vont représenter des relations. Il faut d'abord pouvoir dire des relations *entre quoi* : la première pièce de la construction sémantique définit les entités qui sont les objets de ces relations.

On considère donc un **univers** $\mathcal{U}$ qui va être, en langage naïf, ce dont les fbf parlent. L'univers est un ensemble - donc un objet connu qui a éventuellement une structure. Puis on établit une correspondance entre le langage (la LPO) et l'univers. Cette correspondance s'appelle une **interprétation**. Une interprétation est composée de :

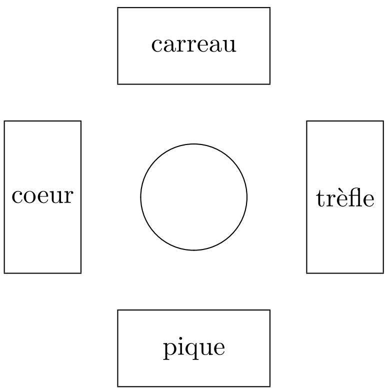
1. une application de l'ensemble des symboles de variables dans $\mathcal{U}$.
2. une application qui fait correspondre à chaque symbole de fonction d'arité n une fonction de $\mathcal{U}^n$ dans $\mathcal{U}$
3. une application qui fait correspondre à chaque symbole de relation d'arité n une partie de $\mathcal{U}^n$.

Comme cas particulier de 2, l'interprétation fait correspondre à une constante un élément de $\mathcal{U}$. L'interprétation s'étend récursivement à tous les termes : si t est un symbole de variable ou de constante, son interprétation $\mathcal{I}$.t est donnée par 1 et 2 ; si t est un terme composé, il s'écrit f(t_1, $\ldots t_n$) et son interprétation est $\mathcal{I}$.f($\mathcal{I}.t_1, \ldots \mathcal{I}.t_n$). Pour alléger la notation, quand o n'a pas besoin de préciser $\mathcal{I}$, on écrira f' au lieu de $\mathcal{I}$.f, t'$_1$ $\ldots$ t'$_n$ au lieu de $\mathcal{I}.t_1, \ldots \mathcal{I}.t_n$.

Remarquez que $t_1 \neq t_2$ n'oblige pas à avoir t'$_1$ $\neq$t'$_2$. Prenons un exemple simple. Voici d'abord le langage

Variables : x
Constantes : a, b, c, d, e
fonctions : f_1, f_2, toutes deux d'arité 1
Relations : R_1, R_2, R_3 d'arité 2

L'univers est un ensemble de 4 cartes (les 4 as) posées aux 4 places d'une table de jeu.



On interprète le langage comme indiqué :

Elements du langage	interprétés par
Variables : x	as de pique
Constantes : a, b, c, d, e	a' = as de trèfle, b'= as de carreau, c' = as de coeur , d'= as de pique, e' = as de trèfle
fonctions : f_1, f_2, toutes deux d'arité 1	f_1' = à gauche de, f_2'= à droite de
Relations : R_1, R_2 d'arité 2	R_1' = vis à vis, R_2' = identité, R_3'=voisin

Il est facile d'énumérer par exemple les 4 couples qui composent R_1'.

Il reste à définir à quelles conditions une formule est valide dans une interprétation. Comme d'habitude, la définition est récursive. Pour une formule atomique, on utilise l'interprétation des termes déja définie ci-dessus. La véritable nouvauté est l'interprétation des quantificateurs.

Définition 8 *$\mathcal{I}$ satisfait F, ou encore F est valide dans $\mathcal{I}$, ou $\mathcal{I}$ est un modèle de F (noté $\mathcal{I} \models F$) si et seulement si l'une des conditions suivante est vérifiée :*

- F est une formule atomique $R(t_1, \ldots, t_n)$ et $(t'_1, \ldots, t'_n) \in R'$
- $F = \neg F'$ et $\mathcal{I} \not\models F'$
- $F = F1 \vee F2$ et soit $\mathcal{I} \models F1$, soit $\mathcal{I} \models F2$
- $F = F1 \wedge F2$ et à la fois $\mathcal{I} \models F1$ et $\mathcal{I} \models F2$
- $F = F1 \rightarrow F2$ et soit $\mathcal{I} \not\models F1$, soit $\mathcal{I} \models F2$
- $F = F1 \leftrightarrow F2$ et soit $\mathcal{I} \models F1$ et $\mathcal{I} \models F2$, soit $\mathcal{I} \models \neg F1$ et $\mathcal{I} \models \neg F2$
- $F = \forall x F_1(x)$ (où x peut être n'importe quelle variable du langage), et **toute interprétation** $\mathcal{I}_1$ qui est identique à $\mathcal{I}$ sauf peut-être pour x satisfait $F_1(x)$.
- $F = \exists x F_1(x)$ (où x peut être n'importe quelle variable du langage), et **il y a au moins une interprétation** $\mathcal{I}_1$ qui est identique à $\mathcal{I}$ sauf peut-être pour x et qui satisfait $F_1(x)$.

Reprenons l'exemple ci-dessus. On vérifie facilement que $\mathcal{I} \models R_1(x,b)$, $\mathcal{I} \models R_2(a,e)$ et $\mathcal{I} \not\models R_2(x,b)$. On vérifie aussi que $\mathcal{I}$ est un modèle de $\forall x.R_1(f_1(f_1(x)), x)$, $\forall x.R_2(f_1(f_1(x)),f_2(f_2(x)))$, $\exists x.R_1(x,e)$, $\exists x.(R_3(x,d) \wedge R_3(x,b))$.

Il reste à donner une dernière définition, celle de la validité :

Définition 9 *La fbf F est valide (ou : est une tautologie) si et seulement si toute interprétation est un modèle de F.*

Il faut quand même repérer deux subtilités dans cette définition. D'abord, l'univers n'est pas précisé. Il s'agit bien de toute interprétation qu'on peut construire, quel qu'en soit l'univers. Ensuite, la définition n'exclut pas que F ait des variables libres. Dans ce cas, si l'on considère une formule close G obtenue en liant toutes les variables libres de F par un quantificateur universel placé en tête, F et G ont les mêmes conditions de validité.

3.4 Modele de Herbrand et tableaux

3.4.1 Modèles de Herbrand

N'importe quel ensemble peut être muni de multiples façons de fonctions et de relations, donc servir d'univers pour des interprétations. On est donc devant une difficulté sérieuse : comment peut-on réussir à montrer qu'une fbf est satisfaite par toutes les interprétations possibles (ou par aucune, ce qui est un problème équivalent). L'avancée la plus importante sur cette question repose sur une idée due à Herbrand : le langage de la LPO est lui aussi un ensemble, et donc il peut servir à construire des interprétations (les interprétations ou les modèles de Herbrand). Or si l'on étudie ces interprétations, on s'aperçoit que résoudre le problème de la satifiabilité dans toutes les interprétations de Herbrand suffit à résoudre le problème général. Voici les grandes lignes de la construction.

On appelle base de Herbrand (notée $\mathcal{B}_\mathcal{H}$) l'ensemble des formules atomiques sans variable. Chaque partie $\mathcal{P}$ de $\mathcal{B}_\mathcal{H}$ définit un modèle de Herbrand ainsi :

- l'univers $\mathcal{U}_\mathcal{H}$ est l'ensemble des termes sans variables (pour avoir un univers non-vide, on suppose qu'il y a au moins une constante et sinon on l'ajoute au langage).
- l'interprétation fait correspondre à chaque symbole de fonction f du langage la fonction f' de ${\mathcal{U}_\mathcal{H}}^n$ dans $\mathcal{U}_\mathcal{H}$ f : $(t_1, \ldots t_n) \mapsto f(t_1', \ldots t_n')$.
- l'interprétation de la relation R est la relation R' de ${\mathcal{U}_\mathcal{H}}^n$: $(t_1, \ldots t_n) \in R'$ ssi $R(t_1, \ldots t_n) \in \mathcal{B}_\mathcal{H}$

Le résultat de Herbrand Si une fbf F a un modèle de Herbrand, elle a bien sûr un modèle. Ce qui est moins trivial, c'est que si elle a un modèle, alors elle a un modèle de Herbrand, sur un langage éventuellement un peu plus riche que le langage d'origine. La construction se fait en 4 temps.

- Toute fbf close est équivalente à une fbf close dont les quantificateurs sont en tête et les négations devant des atomes. On appelle cette forme la forme prenexe.
  Preuve par itération sur la longueur de la formule, en utilisant les règles suivantes :
  – (∀x A ∧ ∀x B) ↔ ∀x (A ∧ B)
  – (∃x A ∨ ∃x B) ↔ ∃x (A ∨ B)
  – (A ∨ ∀x B) ↔ ∀x (A ∨ B) si x n'est pas libre dans A
  – (A ∧ ∃x B) ↔ ∃x (A ∧ B) si x n'est pas libre dans A
  – ∀x A ↔ A si x n'est pas libre dans A
  – ∃x A ↔ A si x n'est pas libre dans A *redondant avec le précédent*
  – ∀x A ↔ ∀y A[x\y] si y n'est pas libre dans A
- une fbf en forme prenexe et qui ne comporte que des quantificateurs universels s'appelle une fbf universelle. Une fbf universelle a un modèle si et seulement si elle a un modèle de Herbrand
  L'idée de la preuve est simple : soit $\mathcal{M}$ un modèle de F. Chaque formule atomique sans variable est soit valide, soit non valide dans $\mathcal{M}$. On prend comme base de Herbrand celles qui sont valides. Il est alors facile de montrer par récurence sur la longueur de la fbf que les mêmes formules sont satisfaites dans $\mathcal{M}$ et dans le modèle de Herbrand ainsi construit.
- Skolemisation : quand la fbf en forme prenexe n'est pas universelle, on la transforme en une fbf universelle (non équivalente !). On applique récursivement la méthode suivante : en partant de la gauche, on repère le premier quantificateur existentiel, soit x_n sa variable et n le nombre de quantificateurs universels qui la précèdent ; on efface $\exists x_n$ de la suite des quantifieurs, on introduit une *nouvelle* fonction f_n d'arité n, et on remplace partout dans le corps de F x_n par $f_n(x_0, \ldots, x_{n-1})$. Quand tous les quantificateurs existentiels ont été éliminés, on a la transformée de Skolem SF.
- dernière étape, si F a un modèle, alors SF a un modèle. Pour chaque valeur du n-uplet $(x_0, \ldots, x_{n-1})$, il suffit de choisir dans le modèle une des valeurs possibles de x_n et d'interpréter $f_n(x_0, \ldots, x_{n-1})$ par cette valeur (en respectant l'ordre de gauche à droite). Le modèle de Herbrand de SF en découle.

Ce qu'il faut comprendre En tant qu'éléments de langage, on ne sait pas si deux termes distincts (a et x dans l'exemple) seront interprétés par deux éléments différents. En tant qu'éléments de l'univers, ils sont distincts. Autre exemple, dans certains modèles d'un langage avec f(), x, y, f(x,y) et f(y,x) peuvent désigner toujours la même chose - par exemple si on interprète x et y par des entiers, f par l'adition. Dans un modèle de Herbrand formé sur ce langage, ce sont toujours des éléments différents de l'univers.

A l'inverse, l'univers d'un modèle quelconque peut être très peuplé - avoir beaucoup plus d'éléments que le langage n'en comporte, et dans ce cas, beaucoup d'éléments du modèle n'ont pas de nom. Le modèle de Herbrand des formules universelles est construit sur un univers d'objet tous nommés. Ce que dit le résultat de Herbrand, c'est que, si le nombre et la taille des fbf sont finis, il suffit d'ajouter un nombre finis de symboles servant à construire des noms pour décider si la formule a un modèle.

Attention toutefois à une nuance : 'un nombre finis de symboles servant à construire des noms' ne signifie pas un nombre fini de noms. Dès qu'il y a au moins un symbole de fonction, le nombre d'objets et donc le nombre d'atomes propositionnels qui déterminent une interprétation est infini. Donc si F a un modèle, on le trouvera en temps fini, mais s'il n'en a pas on ne pourra pas le prouver (au moins, pas par une méthode énumérative).

3.4.2 Tableaux au 1er ordre

C'est la méthode de la section 1.5, mais où les variables sont maintenant des variables de formules du 1er ordre, et les règles sont augmentée pour pouvoir traiter les quantificateurs :

- Si le noeud contient $\neg\neg A$ ajouter un noeud remplaçant cette formule par A
- Si le noeud contient $(A \wedge B)$ ajouter un noeud en remplaçant cette formule par A, B
- Si le noeud contient $\neg(A \vee B)$ ajouter un noeud remplaçant cette formule par $\neg A$, $\neg B$
- Si le noeud contient $\neg(A \rightarrow B)$, ajouter un noeud en remplaçant cette formule par A, $\neg B$.
- Si le noeud contient $\neg(A \wedge B)$ ajouter deux noeuds, un remplaçant cette formule par $\neg A$ et l'autre par $\neg B$
- Si le noeud contient $(A \vee B)$ ajouter deux noeuds, un remplaçant cette formule par A et l'autre par B
- Si le noeud contient $(A \rightarrow B)$ ajouter deux noeuds, un remplaçant cette formule par $\neg A$ et l'autre par B (on pourrait enlever cette règle, puisque $A \rightarrow B$ est $(\neg A) \vee B$; elle fait gagner du temps.
- Si le noeud contient $(A \leftrightarrow B)$ ajouter deux noeuds. Un remplace cette formule par A, B et l'autre par $\neg A$, $\neg B$
- Si le noeud contient $\neg(A \leftrightarrow B)$ ajouter deux noeuds. Un remplace cette formule par A,$\neg B$ et l'autre par $\neg A$, B
- Si le noeud contient $\neg\forall X.A(X)$, introduisez une nouvelle constante de Skolem c et ajouter un noeud en remplaçant cette formule par $\neg A(c)$. Cette règle peut être utilisée plusieurs fois, mais on doit introduire une nouvelle constante de Skolem à chaque fois.
- Si le noeud contient $(\exists X)\ A(X)$, introduisez encore une nouvelle constante de Skolem c et ajoutez un noeud en remplaçant cette formule par A(c)
- Si le noeud contient $(\forall X)\ A(X)$ et t est un terme sans variable, ajoutez un noeud qui contient A(t). On peut faire cette opération pour un nombre quelconque de termes sans variable.
- Si le noeud contient $\neg(\exists X)A(X)$ et t est un terme sans variable quelconque, ajouter un noeud qui contient $\neg A(t)$.

Cours 4

Logique des prédicats 2

4.1 Intuitions

Les intuitions sont fondamentalement les mêmes que pour la présentation syntaxique de la logique propositionnelle : on oublie (provisoirement !) la présentation sémantique, et l'on va décrire des axiomes et des règles d'inférences qui permettent de déduire un certain nombre de fbf, la *clotûre théorique* de ces axiomes[1]

4.2 Axiomes logiques et règles d'inférence.

On conserve tous les axiomes PROP1 à PROP11 du calcul propositionnel, à la différence toutefois qu'ils s'appliquent maintenant à des formules atomiques et non à des proposition atomiques (cf 7.1). On ajoute des axiomes spécifiques notés PRED12, etc. Dans ces axiomes, les formules comportent donc éventuellement des variables, dont une est notée x, et t est un terme qui est substituable à x dans F (par exemple x lui-même).

$\forall x F(x) \rightarrow F(t)$ **(PRED1)** (en particulier, $\forall x F(x) \rightarrow F(x)$)
$F(t) \rightarrow \exists x F(x)$ **(PRED2)** (en particulier, $F(x) \rightarrow \exists x F(x)$)

Dans les schémas PRED3, PRED4 ci-dessous, F est une formule quelconque, et G est une formule dans laquelle x ne figure pas comme variable libre.

$\forall x(G \rightarrow F(x)) \rightarrow (G \rightarrow \forall x F(x))$ **(PRED3)**
$\forall x(F(x) \rightarrow G) \rightarrow (\exists x F(x) \rightarrow G)$ **(PRED4)**

On utilise aussi deux règles d'inférences

Modus Ponens : $P \rightarrow C, P \vdash Q$ **(PRED5)**
Generalization : $F(x) \vdash \forall x F(x)$ **(PRED6)**

4.3 Déduction naturelle

Deux définitions préalables :

- A[x :=t] désigne la formule obtenue en substituant t à chaque occurence libre de x dans A
- Le terme t est librement substituable à x dans A ssi A ne comporte aucune occurence libre de x dans la portée d'un quantificateur liant une variable de t.

Les règles de la logique propositionnelle restent valables. On ajoute pour la manipulation des quantificateurs les règles suivantes :

[1] 'Clotûre théorique' doit se comprendre : c'est une théorie qui est close pour les règles d'inférence

$$\frac{A}{(\forall x.A)}\ \mathrm{I}_{\forall}$$ si x n'est pas libre dans les hypothèses en cours

$$\frac{(\forall x.A)}{A[x := t]}\ \mathrm{E}_{\forall}$$ où t est librement substituable à x dans A

$$\frac{A[x := t]}{(\exists x.A)}\ \mathrm{I}_{\exists}$$ où t est librement substituable à x dans A

$$\frac{(\exists x.A), (A \rightarrow B)}{B}\ \mathrm{E}_{\exists}$$ si x n'est libre ni dans les hypothèses en cours, ni dans B.

4.4 Séquents

Les règle données au cours 2 sont toujours valables. On ajoute 4 règles spécifiques aux quantificateurs. Dans ces règles, on note α une variable libre de la formule $F(\alpha)$, et $F(x)$ le résultat de la substitution de x à α dans F. De plus, dans $\forall$ gauche et $\exists$ droit, α ne doit pas apparaître dans le séquent inférieur (la conclusion).

	gauche	droit
$\forall$	$\dfrac{F(\alpha), \Lambda \vdash \Gamma}{\forall x F(x), \Lambda \vdash \Gamma}$	$\dfrac{\Gamma \vdash F(\alpha)}{\Gamma \vdash \forall x F(x)}$
$\exists$	$\dfrac{F(\alpha), \Lambda \vdash \Gamma}{\exists x F(x), \Lambda \vdash \Gamma}$	$\dfrac{\Gamma \vdash F(\alpha)}{\Gamma \vdash \exists x F(x)}$

4.5 Démonstration automatique

Cours 5

Logique modale 1

5.1 Intuitions

5.1.1 jugements sur une proposition

La logique propositionnelle comme la logique des prédicats décrivent des raisonnements sur des formules en décidant si elles sont vraies ou fausses – mais c'est tout ce qu'elle peut en dire : dans un modèle, on a soit p, soit ¬p. Dans la réflexion quotidienne ou savante sur la réalité (science, philosophie, pédagogie, politique, dialogue Homme-machine, vie quotidienne, . . .), il y a des jugements plus nuancés sur la validité d'une proposition. Par exemple :

Dans la tradition philosophique :

* "Il est 10h" vérité déictique (dépend du moment ou du lieu où on la formule) ;
* "Au 20^{eme} siècle, le régime politique de la France est la république" vérité contingente (il aurait pu en être autrement).
* "La Grèce est bordée par la méditerranée" ? ? ?
* " Le carré de l'hypothénuse est la somme des carrés des deux autres cotés" Vérité nécessaire.

Dans l'usage courant :

* "Il peut pleuvoir cet après midi" éventualité
* "John peut faire 300 km en vélo dans la journée" capacité
* "je dois rendre ma copie demain" obligation
* "je sais que tu es là" connaissance

On peut explorer plusieurs extensions du formalisme de base pour exprimer ces nuances. Une première idée est d'enrichir la liste des valeurs de vérité. Nous ne suivrons pas cette piste, mais elle le sera sans doute dans un autre cours. La seconde idée est de créer de nouveaux types de propositions. On utilise pour cela des **opérateurs modaux**. Les plus courants sont : $\Box$ (nécessaire), $\Diamond$ (possible), **K**. (sait que), **B**. (croit que), **O**. (obligatoire), **A**. (permis).

Par exemple, si p est une proposition, $\Box$p est une nouvelle proposition (nécessaire p), de même que **K**p (sait que p) ou **A**p (autorisé p). Nous allons nous intéresser à toute une série de calculs construits autour d'au moins un de ces opérateurs, et qui essaieront de modéliser les rapports entre par exemple 'je sais que p' et 'p est vrai'. Nous donnerons de ces calculs des définitions syntaxiques et des définitions sémantiques, en nous demandant comme dans les chapitres précédents si les deux coïncident. A chaque fois, deux questions essentielles se posent : ce calcul représente-t-il bien les raisonnements qui nous intéressent ou au moins une partie d'entre eux, et a-t-il des propriétés calculatoires qui le rendent utilisable ?

5.1.2 Formules intéressantes

Avant d'aborder la présentation formelle, nous présentons quelques schémas de formules dont on peut se demander si elles conviennent aux propriétés de telle ou telle modalité, et donc qu'on peut souhaiter ou non inclure dans une logique modale particulière. Chacune a un nom traditionnel, que nous emploierons régulièrement par la suite.

(T)	$\Box A \rightarrow A$	$\mathbf{K}.A \rightarrow A$	$\mathbf{O}.A \rightarrow A$
($T_\Diamond$)	$A \rightarrow \Diamond A$	$A \rightarrow \mathbf{B}.A$	$A \rightarrow \mathbf{A}.A$
($def_\Diamond$)	$\Diamond A \leftrightarrow \neg\Box\neg A$	$\mathbf{B}.A \leftrightarrow \neg\mathbf{K}.\neg A$	$\mathbf{A}.A \leftrightarrow \neg\mathbf{O}.\neg A$
(D)	$\Box A \rightarrow \Diamond A$	$\mathbf{K}.A \rightarrow \mathbf{B}.A$	$\mathbf{O}.A \rightarrow \mathbf{A}.A$
(N)	$\Box(A \vee \neg A)$	$\mathbf{K}.(A \vee \neg A)$	$\mathbf{O}.(A \vee \neg A)$
(K)	$\Box(A\rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$	$\mathbf{K}.(A\rightarrow B) \rightarrow (\mathbf{K}.A \rightarrow \mathbf{K}.B)$	$\mathbf{O}.(A\rightarrow B) \rightarrow (\mathbf{O}.A \rightarrow \mathbf{O}.B)$

5.2 Langage

Le langage de la logique modale propositionnelle est formé à partir d'un ensemble dénombrable P de symboles de propositions : $P = \{p_i \mid i \in I\}$. Comme dans le cas classique, la définition est récursive :

L'ensemble des propositions bien formées (pbf) est le plus petit ensemble qui contient P et tel que :

si A, A1 et A2 sont des pbf, alors
- $\neg A$ est une pbf
- $(A1 \wedge A2)$ est une pbf
- $(A1 \vee A2)$ est une pbf
- $(A1 \rightarrow A2)$ est une pbf
- $(A1 \leftrightarrow A2)$ est une pbf
- $\Box A$ est une pbf
- $\Diamond A$ est une pbf

5.3 Modèles standard : les intuitions

On peut voir un modèle de la logique propositionnelle classique comme décrivant l'état du monde : les propositions y sont vraies ou fausses. Pour rendre compte du fait qu'une proposition peut aussi être possible, nécessaire, ... il faut rafiner un peu plus la description sémantique. C'est ce qu'on appelle la théorie des *mondes possibles*. Un monde reste caractérisé par les propositions qui y sont vraies ou fausses, mais un modèle comporte plusieurs mondes.

Voyons d'abord la version la plus simple, historiquement la première. Quand nous passerons au formalisme, ce sera un cas particulier de la définition générale (on l'appelle le modèle uniersel). On considère la seule modalité "nécessaire". Un modèle est un ensemble de mondes, et une interprétation qui pour chaque symbole propositionnel donne les mondes où il est interprété à vrai (dans les autres, il vaut faux). Pour les pbf sans symbole modal, la construction classique donne leur validité monde par monde, sans tenir compte des autres mondes. Une même proposition peut donc être vraie dans un monde et fausse dans un autre. Dans ce dispositif, on peut faire une différence entre *vrai* et *nécessairement vrai*. Une proposition est nécessairement vraie dans le modèle si elle est vraie dans **tous** les mondes de ce modèle.

Quand on développe les conséquences formelles de cette définition, on se rend compte que c'est une version très simple. On s'en apperçoit très vite à condition de oser la bonne question. Nous rendons l'affirmation "A est nécessairement vraie" par une formule $\Box A$. Celle-ci doit donc recevoir aussi une interprétation. La question est : dans quels mondes la formule $\Box A$ est-elle vraie ? Or la définition ci-dessus donne la même réponse dans tous les mondes. Donc si $\Box$A est vraie dans un monde, il en est de même de $\Box\Box$A, $\Box\Box\Box$A, etc. De plus, pour toute proposition, on a soit $\Box$A est nécessaire dans ce modèle, soit $\neg\Box$A est nécessaire dans ce modèle : le modèle se comporte pour les formules sous la portée du connecteur modal comme un modèle classique pour les formules non modales.

J'ai utilisé pour ces explications intuitives l'expression "vrai dans un monde". Le terme consacré est "valide", dans un monde ou dans le modèle. Une autre façon de voir la simplicité de cette version est de remarquer que A est valide dans le modèle si et seulement si $\Box$A est valide dans un monde (n'importe lequel, mais un seul suffit). Autrement dit, n'importe quel monde a l'information sur tout le modèle, et c'est pour cela que dès qu'une formule est gouvernée par une modalité, elle est évaluée à l'identique dans tous les mondes. Pour dépasser cette limitation, les modèles que nous allons étudier introduisent une *relation d'accessibilité* : un monde n'a d'information que sur certaines parties du modèle, les mondes qui lui sont accessibles. Le mot n'est pas choisi au hasard : l'image est que un monde représente un agent ; depuis le monde où il se tient, il en *voit* d'autres (tous ceux qui lui paraissent possibles), et ce qui lui parait nécessaire est ce qu'il voit *partout.* Mais un agent ne voit pas nécessairement tout ce qui existe. A ce prix, on peut faire une différence entre nécessité dans un monde et validité dans le modèle. La notion de modèle que nous allons définir maintenant est due à Kripke, et est appelée *modèles standard*, par opposition à une notion de modèle un peu plus compliquée due à Hintika.

5.4 Modèles standard

5.4.1 Définition

Définition 10 (modèle) *Un modèle standard du langage modal construit sur P est un triplet* $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$ *dans lequel :*

- $\mathcal{W}$ est un ensemble (l'ensemble des mondes)
- $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$ est une relation sur $\mathcal{W}$ (la relation d'accessibilité)
- $\mathcal{V}$ est une application de P dans $2^{\mathcal{W}}$ (les parties de $\mathcal{W}$) qui au symbole propositionnel p_i fait correspondre le sous-ensemble $\mathcal{V}(p_i)$ de $\mathcal{W}$ (l'ensemble des mondes où p_i est évalué à vrai).

Pour le vocabulaire, notez qu'on ne fait pas la distinction entre modèle et interprétation − il n'y a pas de réelle ambiguïté, et que l'on dit couramment *un monde de* $\mathcal{M}$ au lieu de *un monde de* $\mathcal{W}$. L'important est de définir correctement la notion de validité. C'est une construction un peu plus compliquée qu'en logique propositionnelle, parce qu'on distingue la validité dans un monde du modèle, et la validité dans le modèle.

La validité dans un monde ajoute aux règles classiques celles qui sont spécifiques aux opérateurs modaux.

Définition 11 (validité dans un monde) *Pour toute pbf A et tout modèle* $\mathcal{M}$ *et tout monde* α *du modèle* $\mathcal{M}$ $(\alpha \in \mathcal{W})$*, A est valide dans* α *(noté* $\mathcal{M} \models_\alpha A$*) si et seulement si l'une des conditions suivantes est vérifiée :*

- A = p_i est un atome propositionnel, et $\alpha \in \mathcal{V}(p_i)$
- A = $\neg$A1 et $\mathcal{M} \not\models_\alpha$A1
- A = A1 $\vee$ A2 et soit $\mathcal{M} \models_\alpha$A1, soit $\mathcal{M} \models_\alpha$A2
- A = A1 $\wedge$ A2 et $\mathcal{M} \models_\alpha$A1 et $\mathcal{M} \models_\alpha$A2
- A = A1 $\rightarrow$ A2 et soit $\mathcal{M} \not\models_\alpha$A1, soit $\mathcal{M} \models_\alpha$A2
- A = A1 $\leftrightarrow$ A2 et, ou bien $\mathcal{M} \models_\alpha$A1 et $\mathcal{M} \models_\alpha$A2, ou bien $\mathcal{M} \not\models_\alpha$A1 et $\mathcal{M} \not\models_\alpha$A2
- A = $\Box$A1 et tout monde β qui vérifie $\mathcal{R}(\alpha, \beta)$est tel que $\mathcal{M} \models_\beta$A1
- A = $\Diamond$A1 et il y a au oins un monde β de $\mathcal{M}$ qui vérifie $\mathcal{R}(\alpha, \beta)$et tel que $\mathcal{M} \models_\beta$A1

Seules les deux dernières conditions sont nouvelles.

La validité dans un modèle est simplement la validité dans tous les mondes de ce modèle :

Définition 12 *La fbf A est valide dans le modèle $\mathcal{M}$ (noté $\mathcal{M} \models A$) si et seulement si pour tout monde α de $\mathcal{M}$, on a $\mathcal{M} \models_\alpha A$*

Bien entendu, une formule est universellemment valide, ou est une tautologie, si elle est valide dans tous les modèles.

5.4.2 Premières propriétés

La notion de modèle standard permet bien de faire la distinction qui manquait dans le modèle universel. Une formule peut être nécessaire dans un monde sans être valide dans tous. Par contre, une formule valide dans le modèle est nécessaire dans tous ses mondes.

Certains des schémas de formules que nous avons remarqués au § 5.1.2 sont valides dans tous les modèles standard :

(def$_\Diamond$) $\Diamond A \leftrightarrow \neg\Box\neg A$

La preuve est assez simple :

$\mathcal{M} \models_\alpha \neg\Box\neg A$	ssi	$\mathcal{M} \not\models_\alpha \Box\neg A$
	ssi	$\mathcal{M}$ ne vérifie pas : pour tout β t.q. $\mathcal{R}(\alpha,\beta)$ $\mathcal{M} \models_\beta \neg A$
	ssi	il y a au moins un β t.q. $\mathcal{R}(\alpha,\beta)$ et $\mathcal{M} \not\models_\beta \neg A$
	ssi	il y a au moins un β t.q. $\mathcal{R}(\alpha,\beta)$ et $\mathcal{M} \models_\beta A$
	ssi	$\mathcal{M} \models_\alpha \Diamond A$

Ce qui démontre la validité de $\Diamond A \leftrightarrow \neg\Box\neg A$ dans n'importe quel monde □ qed

(N) $\Box\top$ (où $\top$ est n'importe quelle tautologie propositionnelle, par ex. $p_0 \vee \neg p_0$)

Preuve :	$\mathcal{M} \models_\alpha \Box\top$	ssi	pour tout β t.q. $\mathcal{R}(\alpha,\beta)$ $\mathcal{M} \models_\beta \top$
		ssi	pour tout β t.q. $\mathcal{R}(\alpha,\beta)$, soit $\mathcal{M} \models_\beta p_0$, soit $\mathcal{M} \not\models_\beta p_0$.

Or la dernière proposition est trivialement vraie. □ qed

(K) $\Box(A \to B) \to (\Box A \to \Box B)$

Soit α un monde du modèle $\mathcal{M}$. Il faut montrer que l'une au moins des deux conditions suivantes est satisfaite :

(C1) $\mathcal{M} \not\models_\alpha \Box(A \to B)$
(C2) $\mathcal{M} \models_\alpha (\Box A \to \Box B)$

Supposons que ni (C1) ni (C2) n'est satisfaite en α. On a alors

	$\mathcal{M} \models_\alpha \Box(A \to B)$ (C1 non satisfaite)
donc	pour tout β tel que $\mathcal{R}(\alpha,\beta)$, $\mathcal{M} \models_\beta A \to B$
	$\mathcal{M} \models_\alpha \Box A$ et $\mathcal{M} \not\models_\alpha \Box B$ (C2 non satisfaite)
donc	pour tout β tel que $\mathcal{R}(\alpha,\beta)$, $\mathcal{M} \models_\beta A$ (1^{ere} conséquence de (C2))
et	il y a au moins un β tel que $\mathcal{R}(\alpha,\beta)$ et $\mathcal{M} \not\models_\beta B$ (2^{de} conséquence de (C2))

Soit β_0 un des mondes décrits dans la dernière condition. β_0 satisfait à la fois $A \to B$, A et $\neg B$, ce qui est impossible. Ce qui prouve que au moins une des (C1), (C2) est satisfaite en α. □ qed

Donc dans tous les modèles standard (def$_\Diamond$), (N) et (K) sont valides. L'ensemble des formules valides dans tous les modèles standard s'appelle la logique K. Ce n'est pas la seule logique possible. On peut en effet se restreindre à une partie des modèles standard, en choisissant une classe de modèles[1]

Définition 13 (validité dans une classe de modèles) *On dit qu'une fbf est valide dans une* ***classe*** *C de modèles (on écrit $\models^{\mathsf{C}} A$) si et seulement si A est valide dans tout modèle de C.*

On peut définir **une** logique modale comme l'ensemble des fbf satisfaites par une certaine classe de modèles. Il y a d'autres façons de définir une logique modale (par ex. axiomatique). Une logique modale qui peut être caractérisée par une classe de modèles standard est appelée une logique modale **normale**.

Comme premier exemple (trivial), si Std est la classe de tous les modèles standard, A est dans K si et seulement si elle est satisfaite dans la classe des modèles standard ($\models^{\mathsf{Std}} A$).

[1] La notion de classe est non définie : intuitivement, elle correspond à ce que l'on aurait envie d'appeler un ensemble de modèles – mais il faut éviter le paradoxe de Russel (l'ensemble de tous les ensembles n'est pas un ensemble).

Cours 6

Logique modale 2

6.1 Classes de modèles : un peu d'enthomologie

Nous avons terminé le cours précédent avec la notion de validité dans une classe de modèle. Nous en montrons ici quelques emplois élémentaires.

6.1.1 Modèles universels

Nous revenons sur la classe U des modèles universels, caractérisés par : quels que soient les deux mondes α et β de $\mathcal{W}$, ils vérifient $\mathcal{R}(\alpha, \beta)$. On reconnait l'exemple historique dont nous nous sommes servis au cours précédent, reformulé dans le langage des modèles standard. La logique de U s'appelle $\mathsf{S5}$ ($\mathsf{S5} = \{ \mathrm{A} \mid \models^{\mathsf{U}} \mathrm{A} \}$). Citons quelque schémas de formules valides dans U (donc quelques formules de $\mathsf{S5}$). Les démonstrations sont laissées en exercice :

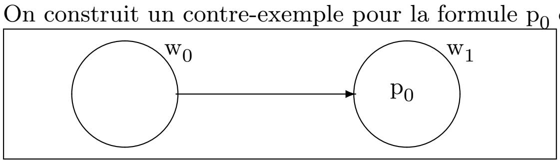
(T)	$\Box \mathrm{A} \rightarrow \mathrm{A}$
(D)	$\Box \mathrm{A} \rightarrow \Diamond \mathrm{A}$
(B)	$\mathrm{A} \rightarrow \Box \Diamond \mathrm{A}$
(4)	$\Box \mathrm{A} \rightarrow \Box\Box \mathrm{A}$
(5)	$\Diamond \mathrm{A} \rightarrow \Box \Diamond \mathrm{A}$

6.1.2 La classe des modèles standard

Nous avons montre ai § 5.4.2 que ($\mathrm{def}_\Diamond$), (N) et (K) sont valides dans tous les modèles standard. Nous reprenons ici les autres formules 'intéressantes' pour montrer qu'elles n'appartiennent pas à la logique K. Ces démonstrations simples permettent d'illustrer l'emploi de contre-exemples.

(T) $\Box \mathrm{A} \rightarrow \mathrm{A}$

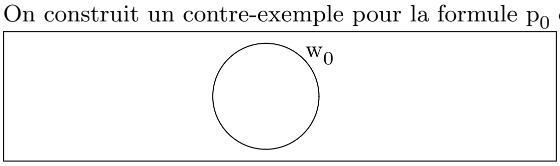
On construit un contre-exemple pour la formule p_0 dans le monde w_0 :



On a $\mathcal{W} \models_{w_0} \Box \mathrm{p}_0$ et $\mathcal{W} \not\models_{w_0} \mathrm{p}_0$, donc $\mathcal{W} \not\models (\Box \mathrm{p}_0 \rightarrow \mathrm{p}_0)$, donc le schéma d'axiome $(\Box \mathrm{A} \rightarrow \mathrm{A})$ n'est pas valide dans Std ($\not\models^{\mathsf{Std}} (\Box \mathrm{A} \rightarrow \mathrm{A})$)

(D) $\Box \mathrm{A} \rightarrow \Diamond \mathrm{A}$

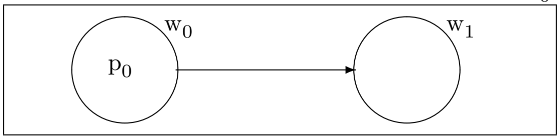
On construit un contre-exemple pour la formule p_0 dans le monde w_0 :



On a $\mathcal{W} \models_{w_0} \Box \mathrm{p}_0$ et $\mathcal{W} \not\models_{w_0} \Diamond \mathrm{p}_0$, donc $\mathcal{W} \not\models (\Box \mathrm{p}_0 \rightarrow \Diamond \mathrm{p}_0)$, donc le schéma d'axiome $(\Box \mathrm{A} \rightarrow \Diamond \mathrm{A})$ n'est pas valide dans Std ($\not\models^{\mathsf{Std}} (\Box \mathrm{A} \rightarrow \Diamond \mathrm{A})$)

(B) $A \rightarrow \Box\Diamond A$

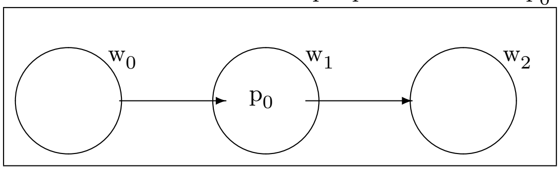
On construit un contre-exemple pour la formule p_0 dans le monde w_0 :



On a $\mathcal{W} \models_{w_0} p_0$ et $\mathcal{W} \not\models_{w_1} \Diamond p_0$, donc $\mathcal{W} \not\models_{w_0} \Box\Diamond p_0$ donc $\mathcal{W} \not\models (p_0 \rightarrow \Box\Diamond p_0)$, donc le schéma d'axiome $(A \rightarrow \Box\Diamond A)$ n'est pas valide dans Std ($\not\models^{\mathsf{Std}} (A \rightarrow \Box\Diamond A)$)

(4) $\Box A \rightarrow \Box\Box A$

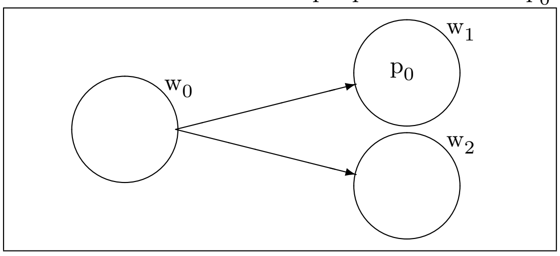
On construit un contre-exemple pour la formule p_0 dans le monde w_0 :



On a d'une part $\mathcal{W} \models_{w_0} \Box p_0$, d'autre part $\mathcal{W} \not\models_{w_2} p_0$ donc $\mathcal{W} \not\models_{w_1} \Box p_0$, donc $\mathcal{W} \not\models_{w_0} \Box\Box p_0$, donc $\mathcal{W} \not\models (\Box p_0 \rightarrow \Box\Box p_0)$, donc le schéma d'axiome $(\Box A \rightarrow \Box\Box A)$ n'est pas valide dans Std ($\not\models^{\mathsf{Std}} (\Box A \rightarrow \Box\Box A)$)

(5) $\Diamond A \rightarrow \Box\Diamond A$

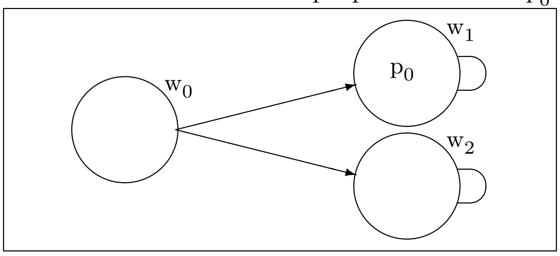
On construit un contre-exemple pour la formule p_0 dans le monde w_0 :



On a d'une part $\mathcal{W} \models_{w_1} p_0$ donc $\mathcal{W} \models_{w_0} \Diamond p_0$, d'autre part $\mathcal{W} \not\models_{w_2} \Diamond p_0$ (monde aveugle) donc $\mathcal{W} \not\models_{w_0} \Box\Diamond p_0$; donc $\mathcal{W} \not\models_{w_0} \Diamond p_0 \rightarrow \Box\Diamond p_0$ donc $\mathcal{W} \not\models (\Diamond p_0 \rightarrow \Box\Diamond p_0)$, donc le schéma d'axiome $(\Diamond A \rightarrow \Box\Diamond A)$ n'est pas valide dans Std ($\not\models^{\mathsf{Std}} (\Diamond A \rightarrow \Box\Diamond A)$)

(G) $\Diamond\Box A \rightarrow \Box\Diamond A$

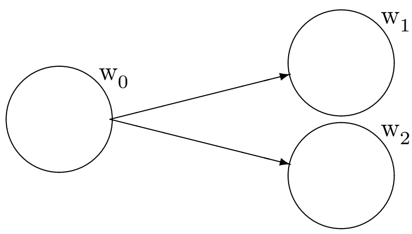
On construit un contre-exemple pour la formule p_0 dans le monde w_0 :



On a d'une part $\mathcal{W} \models_{w_1} \Box p_0$ donc $\mathcal{W} \models_{w_0} \Diamond\Box p_0$, d'autre part $\mathcal{W} \not\models_{w_2} \Diamond p_0$ (w_2 ne 'voit' que $\neg p_0$) donc $\mathcal{W} \not\models_{w_0} \Box\Diamond p_0$; donc $\mathcal{W} \not\models_{w_0} \Diamond\Box p_0 \rightarrow \Box\Diamond p_0$ donc $\mathcal{W} \not\models (\Diamond\Box p_0 \rightarrow \Box\Diamond p_0)$, donc le schéma d'axiome $(\Diamond\Box A \rightarrow \Box\Diamond A)$ n'est pas valide dans Std ($\not\models^{\mathsf{Std}} (\Diamond\Box A \rightarrow \Box\Diamond A)$)

6.2 Cadres et classes

Une façon intéressante de constituer une famille de modèles modaux est de fixer la relation d'accessibilité — tous les modèles de la famille auront la même relation, mais différeront par la valuation choisie pour les lettres propositionnelles. Prenons par exemple la relation utilisée dans le contre-exemple à (5).



Quelle que soit la valuation adoptée, quelle que soit la formule A, $\Box A$ est valide en w_1 et w_2, et donc $\Box\Box A$ est valide en w_0. Et donc finallement $\Box\Box A$ est valide dans $\mathcal{M}$! !

On appelle **cadre** ('frame' en anglais) la donnée d'un ensemble $\mathcal{W}$ de mondes et d'une relation binaire $\mathcal{R}$ sur $\mathcal{W}$. Un modèle $\mathcal{M}$ repose sur le cadre $\mathcal{F}$ si ils ont même ensemble de mondes et même relation d'accessibilité. On dira naturellement qu'une formule est valide dans un cadre ($\mathcal{F} \models$A) si elle est valide dans tous les modèles du cadre. On dira aussi qu'une formule est valide en *un monde* w du *cadre* $\mathcal{F}$ ($\mathcal{F} \models_w$) si elle est valide en w dans tous les modèles reposant sur ce cadre.

Une façon simple de définir une classe de modèle est maintenant de spécifier une propriété du cadre sur lequel reposent les modèles. C'est déja ce que nous avons fait implicitement en considérant la classe des modèles universels : ce sont les modèles dont le cadre comporte une relation universelle. Nous allons voir d'autres exemples en rapport avec les axiomes étudiés dans le paragraphe précédent.

Considérons le schéma d'axiome (T) $\Box$A $\rightarrow$ A. Considérons un monde w où on veut vérifier qu'il est valide. Il s'agit de vérifier que pour toutes les formules A telles que $\Box$A est valide en w, A l'est aussi. Il est trivial de constater que, si $\mathcal{R}$(w,w), c'est bien le cas. Dans les **modèles**, cette condition n'est pas indispensable : cf par ex le schéma 1. Si l'on condidére par contre un monde w d'un **cadre** $\mathcal{F}$ qui ne vérifie pas $\mathcal{R}$(w,w), on construit sans difficulté une valuation $\mathcal{V}$ telle que $\mathcal{F} \models_w \Box p_0$ et $\mathcal{F} \not\models_w p_0$ — donc (T) est valide dans le monde w du **cadre** $\mathcal{F}$ si et seulement si $\mathcal{R}$(w,w). Appliqué à tous les mondes, on obtient :

Propriété 1 *L'axiome (T) caractérise les cadres reflexifs*

Axiomes caractéristiques d'une classe de cadres :

Axiome	schéma de formule	propriété de $\mathcal{R}$	formule en l.p.o. sur $\mathcal{F}$
(T)	$\Box$A $\rightarrow$ A	Reflexive	$\forall$w $\mathcal{R}$(w,w)
(D)	$\Box$A $\rightarrow \Diamond$A	sérielle	$\forall$w$\exists$w' $\mathcal{R}$(w,w')
(4)	$\Box$A $\rightarrow \Box\Box$A	transitive	$\forall$w, w', w" ($\mathcal{R}$(w,w')$\wedge$ $\mathcal{R}$(w',w"))$\rightarrow$ $\mathcal{R}$(w,w")
(5)	$\Diamond$A $\rightarrow \Box\Diamond$A	euclidienne	$\forall$w, w', w" ($\mathcal{R}$(w,w')$\wedge$ $\mathcal{R}$(w,w"))$\rightarrow$ $\mathcal{R}$(w',w")
(B)	A $\rightarrow \Box\Diamond$A	symétrique	$\forall$w,w' $\mathcal{R}$(w,w')$\rightarrow$ $\mathcal{R}$(w',w)
(G)	$\Diamond\Box$A $\rightarrow \Box\Diamond$A	incestualité	$\forall$w, w',w" $\exists$w"' ($\mathcal{R}$(w,w')$\wedge$ $\mathcal{R}$(w,w")) $\rightarrow$ ($\mathcal{R}$(w',w"')$\wedge$ $\mathcal{R}$(w",w"'))

Notez que certaines propriétés des cadres ne peuvent pas être caractérisées par un schéma d'axiome modal. C'est le cas par exemple de $\mathcal{R}$ est irreflexive ($\forall$w $\neg\mathcal{R}$(w,w)), intransitive ($\forall$w, w', w" ($\mathcal{R}$(w,w') $\wedge$ $\mathcal{R}$(w',w")) $\rightarrow \neg\mathcal{R}$(w,w")), asymétrique ($\forall$w,w' $\mathcal{R}$(w,w') $\rightarrow \neg\mathcal{R}$(w',w)).

6.3 Méthode des tableaux

La méthode des tableaux que nous avons vue dans la partie sémantique de la LP et de la LPO peut être adaptée à la logique modale – ou plutôt aux logiques modales puisque l'adaptation dépend des axiomes proprement modaux adoptés. Il y a pour l'essentiel deux présentations de cette méthode, avec et sans étiquette explicite. Nous privilégions la première, pédagogiquement plus claire.

6.3.1 Modèle engendré par un monde

Considérons un modèle modal $\mathcal{M}= \langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$ et un monde w où l'on veut évaluer uneformule A. Pour évaluer des sous-formules sous un opérateur modal, il faudra les évaluer dans les mondes accessibles depuis w. Si celles-ci contiennent à leur tour des opérateurs modaux, il faudra réitérer la recherche des mondes accessibles depuis celui où on évalue. Mais quelle que soit A, ce processus ne peut accéder qu'à une partie bien définie de $\mathcal{W}$(dépendant de w bien sur).

Définition 14 (modèle engendré) *Le modèle* $\mathcal{M}_w$ *engendré par w est* $\mathcal{M}_w = \langle \mathcal{W}_w, \mathcal{R}_w, \mathcal{V}_w \rangle$ *tel que :*

$\mathcal{W}_{\mathrm{w}}$ est le plus petit sous-ensemble de $\mathcal{W}$ contenant w et fermé par $\mathcal{R}$ (si v $\in \mathcal{W}_{\mathrm{w}}$ et $\mathcal{R}(v, v')$, alors v' $\in \mathcal{W}_{\mathrm{w}}$

$\mathcal{R}_{\mathrm{w}}$ est la restriction de $\mathcal{R}$ à $\mathcal{W}_{\mathrm{w}}$

$\mathcal{V}_{\mathrm{w}}$ est définie par $\mathcal{V}_{\mathrm{w}}(\mathrm{p}_i) = \mathcal{V}(\mathrm{p}_i) \cap \mathcal{W}_{\mathrm{w}}$

La propriété fondamentale est donc : quelle que soit la formule A, on a $\mathcal{M} \models_w$A si et seulement si $\mathcal{M}_{\mathrm{w}} \models_w$ A.
Idée de démonstration : on note $gen_0(w) = \{$w$\}$ et $gen_{i+1}(w) = \{\beta \mid \mathcal{R}(gen_i(w), \beta)\}$. Montrer par récursivité sur la profondeur d'enchassement e des opérateurs modaux que l'évaluation de A n'utilise que $gen_e(w)$.

6.3.2 Tableaux pour K

L'idée de la méthode des tableaux est de répondre à la question “A est-il satisfiable en w”. Pour cela, on construit les mondes et les valuations partielles qui permettent de répondre, en raisonnant par cas pour la disjonction (comme dans les chapitres précédents, on a donc un arbre de cas). Chaque monde est étiqueté par un nom, et les noms sont construits de telle façon qu'on y lit la relation d'accessibilité.

Les étiquettes sont donc des suites de nombres $\langle l, m, n \ldots \rangle$ de longueur variable. $\langle 1 \rangle$ est w, et au minimum $\langle l, m, n \rangle$ est accessible depuis $\langle l, m \rangle$. λ est une variable d'étiquette, et λ,n désigne l'étiquette formée par λ suivi de n. On exprime les règles de constucton de l'arbre avec la notation suivante :

$$\frac{etiq : \ form}{\begin{array}{c} etiq'_0 : \ form'_0 \\ etiq'_1 : \ form'_1 \end{array}} \qquad \text{et} \qquad \frac{etiq : \ form}{etiq'_0 : \ form'_0 | etiq'_1 : \ form'_1 | \ldots}$$

Les formules empilées ou séparées par un point-virgule sont ajoutées dans le même noeud. La barre verticale indique le choix, donc la création de plusieurs noeuds fils correspondant à des valuations différentes. Un noeud représente toutes les formules sur la branche depis la racine jusqu'à lui. Un Noeud où aucune règle ne produit une nouvelle formule est terminé. Un noeud qui représente à la fois A et ¬A est clos. Un premier groupe de règles reprend celles de la logique propositionnelle :

$$\frac{\lambda : \ \neg\neg P}{\lambda : \ P} \qquad \frac{\lambda : \ P \wedge \ Q}{\lambda : \ P ; Q} \qquad \frac{\lambda : \ \neg(P \vee \ Q)}{\lambda : \ \neg P ; \neg Q}$$

$$\frac{\lambda : \ \neg(P \rightarrow \ Q)}{\lambda : \ P ; \neg Q} \qquad \frac{\lambda : \ \neg(P \wedge \ Q)}{\lambda : \ \neg P | \lambda : \ \neg Q} \qquad \frac{\lambda : \ (P \vee \ Q)}{\lambda : \ P | \lambda : \ Q}$$

$$\frac{\lambda : \ (P \rightarrow \ Q)}{\lambda : \ \neg P | \lambda : \ Q} \qquad \frac{\lambda : \ (P \leftrightarrow \ Q)}{\lambda : \ P ; Q | \lambda : \ \neg P ; \neg Q} \qquad \frac{\lambda : \ \neg(P \leftrightarrow \ Q)}{\lambda : \ P ; \neg Q | \lambda : \ \neg P | Q}$$

Les règles qui suivent sont les vraies innovations, servant à traiter les opérateurs modaux. Deux définitions d'abord :
Une étiquette est **non-contraint** sur une branche si ce n'est un segmment initial d'aucune étiquette sur cette branche. Il est **utilisé** sur la branche si c'est l'étiquette d'un noeud de la branche.
Les règles sont les suivantes :

$\frac{\lambda : \ \Box P}{\lambda, n : \ P}$ Pout tout λ, n utilisé sur la branche	$\frac{\lambda : \ \neg\Box P}{\lambda, n : \ \neg P}$ Pour un λ, n non contraint sur la branche.	$\frac{\lambda : \ \Diamond P}{\lambda, n : \ P}$ Pour un λ, n non contraint sur la branche.	$\frac{\lambda : \ \neg\Diamond P}{\lambda, n : \ \neg P}$ Pout tout λ, n utilisé sur la branche

Si l'on veut montrer que (p ∨ q) → (□p ∨ □q) est **satisfiable**, on construira le tableau suivant :

$$\begin{array}{|c|cc|} \multicolumn{3}{|c|}{\langle 1 \rangle \ (\mathrm{p} \vee \mathrm{q}) \rightarrow (\Box\mathrm{p} \vee \Box\mathrm{q})} \\ \langle 1 \rangle \ \neg(\mathrm{p} \vee \mathrm{q}) & \multicolumn{2}{c|}{\langle 1 \rangle \ (\Box\mathrm{p} \vee \Box\mathrm{q})} \\ \langle 1 \rangle \ \neg\mathrm{p} \, ; \neg\mathrm{q} & \multicolumn{1}{c|}{\langle 1 \rangle \ \Box\mathrm{p}} & \langle 1 \rangle \ \Box\mathrm{q} \\ \text{Terminé} & \multicolumn{1}{c|}{\text{Terminé}} & \text{Terminé} \end{array}$$

Les trois noeuds terminés donnent trois solutions suffisantes. La troisiemme dit qu'un monde aveugle w satisfait la formule.

Si l'on veut montrer que (□p ∨ □q) → □(p ∨ q) est **une tautologie**, on essayera de montrer que sa négation n'est pas satisfiable.

$$\begin{array}{|c|c|} \multicolumn{2}{|c|}{\langle 1 \rangle \ \neg((\Box\mathrm{p} \vee \Box\mathrm{q}) \rightarrow \Box(\mathrm{p} \vee \mathrm{q}))} \\ \multicolumn{2}{|c|}{(\langle 1 \rangle \ \Box\mathrm{p} \vee \Box\mathrm{q}) \, ; \neg\Box(\mathrm{p} \vee \mathrm{q})} \\ \langle 1 \rangle \ \Box\mathrm{p} \, ; \neg\Box(\mathrm{p} \vee \mathrm{q}) & \langle 1 \rangle \ \Box\mathrm{q} \, ; \neg\Box(\mathrm{p} \vee \mathrm{q}) \\ \langle 1, 1 \rangle \ \neg(\mathrm{p} \vee \mathrm{q}) & \langle 1, 1 \rangle \ \neg(\mathrm{p} \vee \mathrm{q}) \\ \langle 1, 1 \rangle \ \neg\mathrm{p} \, ; \neg\mathrm{q} & \langle 1, 1 \rangle \ \neg\mathrm{p} \, ; \neg\mathrm{q} \\ \langle 1, 1 \rangle \ \mathrm{p} & \langle 1, 1 \rangle \ \mathrm{q} \\ \text{clos} & \text{clos} \end{array}$$

On remarquera que l'ordre d'application des règles est important : il faut appliquer ¬□ ou ◇ (qui créent de nouveaux noeuds) avant □ et ¬◇ (qui modifient les noeuds existant) pour éviter les retours en arrière.

6.3.3 Autres logiques

L'idée de l'extension à d'autres logiques modales est la suivante : quans on écrit une règle, λ, n dans la conclusion représente un monde accessible depuis λ. Si l'on peut étendre la règle d'étiquetage pour rendre compte de la classe de cadres caractéristique de cette logique, on raisonnera correctement sur les opérateurs modaux de cette logique. Par exemple, pour (T) caractérisée par les cadres réflexifs on considérera que λ et λ, n sont accessibles depuis λ.

Nous donnons ici les relations d'accessibilité attachées aux logiques les plus courantes (d'après [Fitting 93]).

Définition 15 *Les conditions d'accessibilités sur les étiquettes sont*

La condition générale λ, n est accessible de λ

La condition inverse λ est accessible de λ, n

La condition de reflexivité λ est accessible de lui-même

La condition de transitivité λ est accessible de n'importe lequel de ses segments initiaux

La condition universelle n'importe quelle étiquette est accessible de n'imorte quelle étiquette.

Pour chacune de ces logiques courantes, sa condition d'accessibilité est donnée dans le tableau suivant (En seconde colonne, les formules dont cette logique est conséquence sémantique) :

Logique	Modèle standard +	condition d'accessibilité
K		Générale
D	(D)	Générale
T	(T)	Générale, reflexivité
KB	(B)	Générale, inverse
DB	(D) (B)	Générale, inverse
B	(B) (T)	générale, reflexivité, inverse
K4	(4)	Générale, transitivité
D4	(D) (4)	Générale, transitivité
S4	(T) (4)	Générale, reflexivité, transitivité
S5	(T) (4) (5)	Universelle

Pour K, KB et K4, les règles sont :

$\dfrac{\lambda :\ \Box P}{\tau :\ P}$ Pout tout τ accessible utilisé sur la branche	$\dfrac{\lambda :\ \neg\Box P}{\lambda, n :\ \neg P}$ Pour un λ, n non contraint sur la branche.	$\dfrac{\lambda :\ \Diamond P}{\lambda, n :\ P}$ Pour un λ, n non contraint sur la branche.	$\dfrac{\lambda :\ \neg\Diamond P}{\tau :\ \neg P}$ Pout tout τ accessible utilisé sur la branche

Pour D, T, DB, B, D4, S4, S5

$\dfrac{\lambda :\ \Box P}{\tau :\ P}$ Pour tout τ accessible utilisé sur la branche ou sinon pour un λ, n non contraint sur la branche.	$\dfrac{\lambda :\ \neg\Box P}{\lambda, n :\ \neg P}$ Pour un λ, n non contraint sur la branche.	$\dfrac{\lambda :\ \Diamond P}{\lambda, n :\ P}$ Pour un λ, n non contraint sur la branche.	$\dfrac{\lambda :\ \neg\Diamond P}{\tau :\ \neg P}$ Pour tout τ accessible utilisé sur la branche ou sinon pour un λ, n non contraint sur la branche.

Cours 7

Logique modale 3

Nous allons maintenant nous préoccuper de la présentation syntaxique des logiques modales. Le langage a déja été défini, et nous en restons pour ce chapitre aux méthodes valables dans toutes les logiques modales normales (i.e. ayant des modèles standard).

7.1 Axiomes et règles

Pour axiomatiser le système K, qui est la logique de la classe des modèles standard, on conserve les schémas d'axiomes de la logique propositionnelle (à la différence près qu'on peut y substituer un marqueur de place par une proposition comportant des opérateurs modaux) et la règle d'inférence du Modus Ponens. On y ajoute deux schémas d'axiomes et une règle d'inférence :

(def$_\Diamond$) $\Diamond$A $\leftrightarrow \neg\Box\neg$A

(K) $\Box$(A $\rightarrow$ B) $\rightarrow$ ($\Box$A $\rightarrow$ $\Box$B)

(Nec) $\dfrac{A}{\Box A}$ (i.e. is $\vdash_K$A alors $\vdash_K\Box$A

Cette axiomatique se prète à des démonstrations à la Hilbert, où chaque ligne est soit une instance d'un schéma d'axiome, soit l'application à des lignes précédentes de (Nec) ou (MP). On peut ajouter quelques règles d'inférences dérivées pour raccourcir les preuves, et quelques théorèmes utiles.

(RM) $\dfrac{A \rightarrow B}{\Box A \rightarrow \Box B}$ (monotonie)

preuve : par hypothèse, on a montré A $\rightarrow$ B qui est donc un théorème de K. En utilisant (Nec) on obtient $\Box$(A $\rightarrow$ B). L'axiome (K) s'écrit $\Box$(A $\rightarrow$ B) $\rightarrow$ ($\Box$A $\rightarrow$ $\Box$B). En utilisant (MP), on obtient ($\Box$A $\rightarrow$ $\Box$B) qui est donc aussi un théorème de K.

(RR) $\dfrac{(A \wedge B) \rightarrow C}{(\Box A \wedge \Box B) \rightarrow \Box C}$ (régularité)

preuve : on peut ajouter dans toute démonstration dans (K), aprés A $\wedge$ B $\rightarrow$ C les lignes suivantes :

n.	A $\wedge$ B $\rightarrow$ C	
n+1.	A $\rightarrow$ (B $\rightarrow$ C)	n, LP
n+2.	$\Box$A $\rightarrow$ $\Box$(B $\rightarrow$ C)	n+1, (RM)
n+3.	$\Box$(B $\rightarrow$ C) $\rightarrow$ ($\Box$B $\rightarrow$ $\Box$C)	(K)
n+4.	$\Box$A $\rightarrow$ ($\Box$B $\rightarrow$ $\Box$C)	n+2, n+3, LP (plusieurs lignes)
n+5.	($\Box$A $\wedge$ $\Box$B) $\rightarrow$ $\Box$C	n+4, LP

(RE) $\dfrac{A \leftrightarrow B}{\Box A \leftrightarrow \Box B}$ (equivalence)

preuve : on applique deux fois (RM)

(RK) $\dfrac{(A_1 \wedge \ldots \wedge A_n) \rightarrow A}{(\Box A_1 \wedge \ldots \wedge \Box A_n) \rightarrow \Box A}$ pour tout n$\geq$0

preuve : les cas n=0, 1 et 2 correspondent à (Nec), (RM), (RR). L'induction se fait sur le modèle de (RR), en utilisant l'équivalence de $(A_1 \wedge \ldots \wedge A_n) \rightarrow A$ et de $(A_1 \wedge \ldots \wedge A_{n-1}) \rightarrow (A_n \rightarrow A)$

(N) $\Box\top$

preuve : $\top$ est un théorème, donc par (Nec) $\Box\top$ aussi

(M) $\Box(A \wedge B) \to (\Box A \wedge \Box B)$

1.	$A \wedge B \to A$	PL
2.	$\Box(A \wedge B) \to \Box A$	1., (RM)
3.	$A \wedge B \to B$	PL
4.	$\Box(A \wedge B) \to \Box B$	3., (RM)
5.	$\Box(A \wedge B) \to (\Box A \wedge \Box B)$	2, 4, PL

(C) $(\Box A \wedge \Box B) \to \Box(A \wedge B)$

1.	$(A \wedge B) \to (A \wedge B)$	
2.	$(\Box A \wedge \Box B) \to \Box(A \wedge B)$	1, (RR)

(R) $\Box(A \wedge B) \leftrightarrow (\Box A \wedge \Box B)$

C'est une conséquence immédiate des deux précédentes.

7.2 Déduction naturelle

En déduction naturelle, nous allons raisonner dans un monde w. Il y aura maintenant deux sortes de raisonnements enchassés : les raisonnement sous hypathèse, et les raisonnements dans les mondes accessibles. On indiquera cet enchassement par des cadres

7.2.1 règles non modales

Les règles que nous indiquons ici ne sont qu'une reformulation de celles indiquées pour la logique propositionelle. Nous en donnons une version réduite en supposant que $\wedge$, $\vee$, $\leftrightarrow$ sont définis à partir de $\neg$et $\to$ .Il est facile d'ajouter des règles déduites. Une série de règles permettent d'ajouter des lignes à la démonstration. On représente ces règles en séparant la ligne ajoutée par un trait.

$$\frac{}{\top} \qquad\qquad \frac{\bot}{X} \qquad\qquad \frac{\begin{array}{c} X \\ \neg X \end{array}}{\bot} \qquad\qquad \frac{\begin{array}{c} X \\ X \to Y \end{array}}{Y}$$

Nous indiquons maintenant le raisonnement sous hypothèse par un cadre faible. On peut reprendre à l'intérieur d'un cadre une ligne antérieure, *à condition de ne pas avoir fermé le cadre de la ligne que l'on reprend.* Nous aurons trois règles :

$$\begin{array}{c} \boxed{\begin{array}{c} X \\ \vdots \\ Y \end{array}} \\ X \to Y \end{array} \qquad\qquad \begin{array}{c} \boxed{\begin{array}{c} X \\ \vdots \\ \bot \end{array}} \\ \neg X \end{array} \qquad\qquad \begin{array}{c} \boxed{\begin{array}{c} \neg X \\ \vdots \\ \bot \end{array}} \\ X \end{array}$$

(Décharge d'hypothèse) — (Raisonnement par l'absurde) — (Raisonnement par l'absurde)

7.2.2 Raisonnement modal dans$\mathsf{K}$

Enfin, nous indiquerons le raisonnement sous opérateur modal par un cadre strict Dans K, seules deux règles sons utilisées :

Itération stricte Si $\Box X$ figure avant un cadre strict et **au même niveau** que celui-ci, on peut insérer X dans le cadre strict. Deux items sont au même niveau si ils sont exactement dans les mêmes cadres stricts.

Fermeture du cadre strict On peut fermer un cadre strict après n'importe quelle ligne. Si celle-ci est X, on peut ajouter $\Box X$ après le cadre.

On peut résumer ces règles par :

$$\begin{array}{c} \Box X \\ \Box Y \\ \hline \boxed{\begin{array}{c} X \\ Y \\ \vdots \\ Z \end{array}} \\ \hline \Box Z \end{array}$$

Raisonnement sur $\Box$

Notez que la règle de nécessitation est traitée implicitement, parce que une tautologie peut être introduite dans un cadre strict. L'opérateur $\Diamond$ est considéré comme défini et n'est pas traité directement.

Exemple : montrer $(\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$

$$\begin{array}{c} \boxed{\begin{array}{c} \Box A \wedge \Box B \\ \Box A \\ \Box B \\ \hline \boxed{\begin{array}{c} A \\ B \\ A \wedge B \end{array}} \\ \hline \Box(A \wedge B) \end{array}} \\ (\Box A \wedge \Box B) \rightarrow \Box(A \wedge B) \end{array}$$

7.2.3 Déduction naturelle dans d'autres logiques

On va traduire en règles les axiomes spécifiques à chaque logique. Ces règles sont soit une augmentation de la règle d'itération stricte, soit des règles spécifiques. Comme dans le cas des tableaux, on passe par une forme générale de la règle d'itération stricte :

Itération stricte Soit S l'ensemble des formules qui figurent avant un cadre strict et **au même niveau** que celui-ci, on peut insérer dans le cadre strict n'importe quel membre de l'ensemble $S^{\rightarrow}$ décrit dans la table ci-dessous (la *logique* B est caractérisée par les *axiomes* (K), (B), (T).

Logique	$S^{\rightarrow}$
K, T, D	$\{X \mid \Box X \in S\}$
K4, D4	$\{X \mid \Box X \in S\} \cup \{\Box X \mid \Box X \in S\}$
S4	$\{\Box X \mid \Box X \in S\}$
KB, DB, B	$\{X \mid \Box X \in S\} \cup \{\neg\Box X \mid \neg X \in S\}$
S5	$\{\Box X \mid \Box X \in S\} \cup \{\neg\Box X \mid \neg\Box X \in S\}$

Règles aditionnelles Pour chaque logique, on ajoute la règle d'inférence spécifique ci-dessous :

Logique	règle d'inférence spécifique
K,K4, KB	néant
B, T, S4, S5	$\dfrac{\Box X}{X}$
D, DB, D4	$\dfrac{\Box X}{\neg\Box\neg X}$

7.3 Dualité

La dualité est une technique syntaxique pour générer plus rapidement de nouveaux théorèmes à partir de théorèmes connus, en raccourcissant des démonstrations fastidieuses. Pour la présenter, nous nous appuierons systématiquement sur la technique de substitution de toutes les occurences d'une variable propositionnelle par une même proposition bien formée. On obtient ainsi un nouveau théorème.

Par exemple, si $(\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$ est un théorème, $(\Box(X \rightarrow Y) \wedge \Box(Y \rightarrow Z)) \rightarrow \Box((X \rightarrow Y) \wedge (Y \rightarrow Z))$ est aussi un théorème, puisqu'on a substitué A par $X \rightarrow Y$ et B par $Y \rightarrow Z$. **Attention**, l'inverse est incorrect : la substitution d'une formule (même de $\neg A$) par une autre formule ne préserve pas le fait d'être un théorème.

Que se passe-t-il si dans la formule précédente nous substituons chaque lettre propositionnelle par sa négation ? Nous obtenons successivement

$(\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$

$(\Box\neg A \wedge \Box\neg B) \rightarrow \Box(\neg A \wedge \neg B)$ (substitution)

$(\neg\Diamond A \wedge \neg\Diamond B) \rightarrow \Box\neg(A \vee B)$ ($\Box\neg = \neg\Diamond$ + de Morgan)

$\neg(\Diamond A \vee \Diamond B) \rightarrow \neg\Diamond(A \vee B)$ (de Morgan + $\Box\neg = \neg\Diamond$)

$\Diamond(A \vee B) \rightarrow (\Diamond A \vee \Diamond B)$ (contraposée)

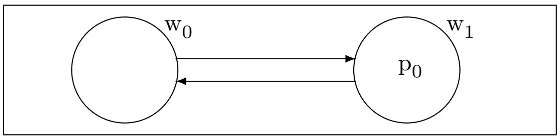
Dans la dernière formule, l'opérateur de nécessité a été remplacé par un opérateur de possibilité. On peut faire cette transformation de façon systématique. La nouvelle formule que l'on obtient s'appelle la formule **duale** de la formule originelle.

Cours 8

Logique modale 4

8.1 Combinaisons d'axiomes et systèmes

Tous les axiomes que nous avons vus ne sont pas indépendants les uns des autres. Nous démontrons quelques propriétés.

- (T) $\rightarrow$ (D)
  (T) est $\Box$A $\rightarrow$ A, (D) est $\Box$A $\rightarrow$ $\Diamond$A. On utilise (def$_\Diamond$) pour montrer que (T) $\rightarrow$ (T$_\Diamond$) avec (T$_\Diamond$) = A $\rightarrow$ $\Diamond$A. Donc par LP $\Box$A $\rightarrow$ $\Diamond$A.
  On conclut que $\mathsf{K} \subset \mathsf{KD} \subset \mathsf{KT}$. L'inclusion est stricte parce que (D) $\not\rightarrow$ (T), comme le prouve le contre-exemple suivant :
  

  On a $\mathcal{W} \models_{w_0} \Box p_0$ et $\mathcal{W} \models_{w_0} \Diamond p_0$, donc $\mathcal{W} \models_{w_0} (\Box p_0 \rightarrow \Diamond p_0)$, mais $\mathcal{W} \models_{w_0} \neg p_0$ donc $\mathcal{W} \not\models_{w_0} (\Box p_0 \rightarrow p_0)$, $\Box$ qed
- (K) + (B) + (4) $\leftrightarrow$ (K) + (B) + (5)
  (K) est $\Box$(A $\rightarrow$ B) $\rightarrow$ ($\Box$A $\rightarrow$ $\Box$B), (B) est A $\rightarrow$ $\Box\Diamond$A (4) est $\Box$A $\rightarrow$ $\Box\Box$A, (5) est $\Diamond$A $\rightarrow$ $\Box\Diamond$A
  On peut faire la démonstration en utilisant les cadres qui caractérisent chacune de ces formules. (K) + (B) + (4) est caractérisée par un cadre symétrique et transitif, (K) + (B) + (5) par un cadre symétrique et euclidien. Montrer l'équivalence de ces deux conditions est laissé en exercice.
- (T) + (5) $\rightarrow$ (B)
  (T) implique (T$_\Diamond$) A $\rightarrow$ $\Diamond$A, (5) est $\Diamond$A $\rightarrow$ $\Box\Diamond$A ; par LP on obtient A $\rightarrow$ $\Box\Diamond$A, qui est (B).
  En utilisant les cadres, la démonstration serait Reflexif + Euclidien implique Symétrique.
- (K) + (T) + (B) + (4) $\leftrightarrow$ (K) + (T) + (B) + (5) $\leftrightarrow$ (K) + (T) + (4) + (5)
  La première équivalence d'après (K) + (B) + (4) $\leftrightarrow$ (K) + (B) + (5). Elle a pour conséquence que (K) + (T) + (B) + (4) implique (5), donc (K) + (T) + (4) + (5)
  Dans l'autre sens, on a (T) + (5) $\rightarrow$ (B), donc (K) + (T) + (4) + (5) $\rightarrow$ (B), ce qui donne (K) + (T) + (4) + (5) $\rightarrow$ (K) + (T) + (B) + (5). On a aussi en passant (K) + (T) + (5) $\leftrightarrow$ (K) + (T) + (B) + (5)

La hiérarchie des quelques systèmes normaux cités est donc représentées dans la figure 8.1 (une flèche d'un système vers un autre indique que les tautologies du premier sont des tautologies du second).

8.2 Modalités distinctes : premiers cas

On appelle modalité une suite d'opérateurs modaux. Deux modalités m_1 et m_2 sont logiquement équivale ntes dans le système SS si $\models^{\mathsf{SS}} m_1 A \leftrightarrow m_2 A$.

8.2.1 Modalités de S5

S5 est la logique des cadres universels, dont nous avions déja remarqué qu'elle est particulièrement simple (cf 6, exercice 2). Toutes les modalités se réduisent à une modalité de longueur au plus un :

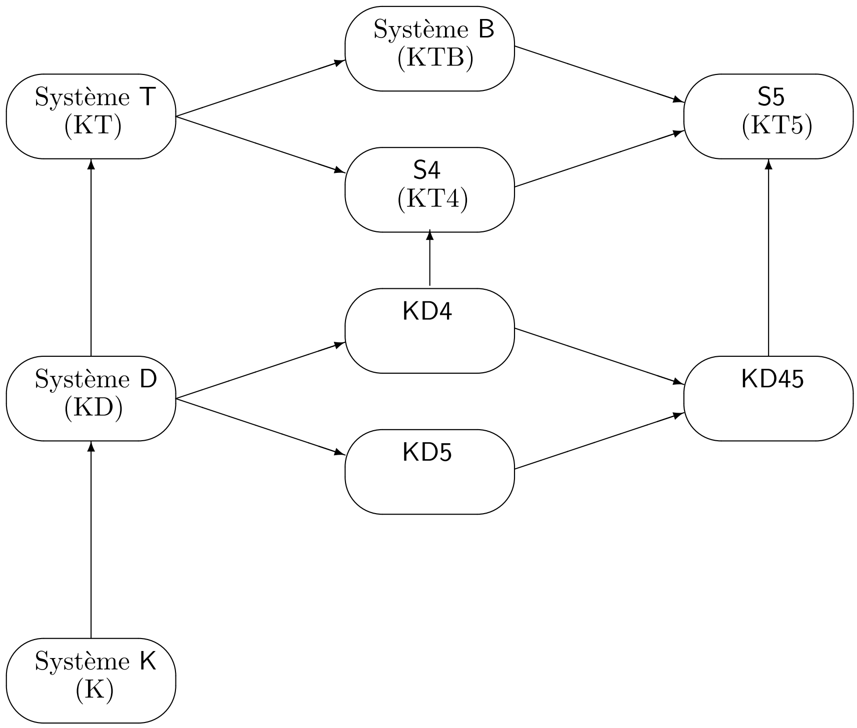
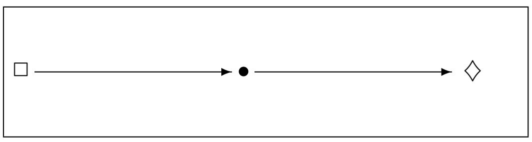


FIG. 8.1 – Hiérarchie des systèmes normaux

$\Box A$	$\leftrightarrow \Box\Box A$	$\rightarrow$: par (4)
		$\leftarrow$: par (T) appliqué à $\Box A$
$\Diamond A$	$\leftrightarrow \Diamond\Diamond A$	appliquer la formule précédente à $\neg A$, puis utiliser (def$_\Diamond$)
$\Diamond A$	$\leftrightarrow \Box\Diamond A$	$\rightarrow$: par (5)
		$\leftarrow$: (T) appliqué à $\Diamond A$
$\Box A$	$\leftrightarrow \Diamond\Box A$	appliquer la formule précédente à $\neg A$, puis utiliser (def$_\Diamond$)

On peut résumer ces résultats simplement : dans une chaine non vide d'opérateurs modaux, seul le dernier compte. Les seules modalisations logiquement distinctes de A sont : A, $\neg A$, $\Box A$, $\Diamond A$, $\Box\neg A$, $\Diamond\neg A$. En ne s'intéressant qu'aux modalités positives, on peut résumer les résultats dans le schéma suivant :



8.2.2 Modalités de **S4**

Nous regardons maintenant ce que deviennent les modalités quand on n'a plus l'axiome (5) d'Euclidianité. L'axiome (B) qui était conséquence de (T) + (5) ne figure plus non plus – avec (4), il ammènerait (5). On est donc dans (S4) = (KT4).

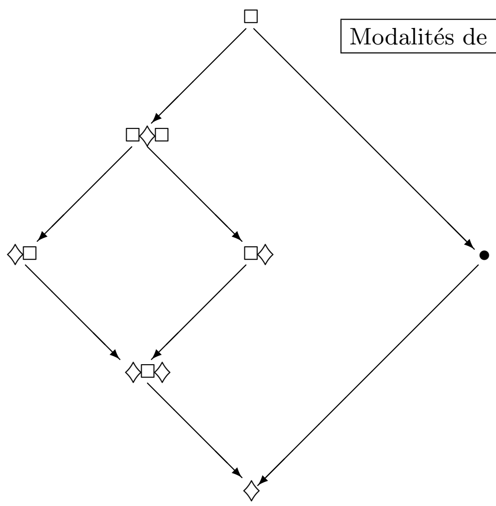
La démonstration de $\Box A \leftrightarrow \Box\Box A$ et $\Diamond A \leftrightarrow \Diamond\Diamond A$ faite au §précédent demeure valide (elle n'utilise que (4) et (T)). On montre aussi (RS4) $\Diamond\Box A \leftrightarrow \Diamond\Box\Diamond\Box A$.

$\rightarrow$	1.	$\Box A \rightarrow \Diamond\Box A$	(T$_\Diamond$)
	2.	$\Box\Box A \rightarrow \Box\Diamond\Box A$	1 par (RM)

	3.	$\Box A \rightarrow \Box\Box A$	(4)
	4.	$\Box A \rightarrow \Box\Diamond\Box A$	2 + 3
	5.	$\Diamond\Box A \rightarrow \Diamond\Box\Diamond\Box A$	4 par ($RM_\Diamond$)
$\leftarrow$	1.	$\Box\Diamond\Box A \rightarrow \Diamond\Box A$	(T)
	2.	$\Diamond\Box\Diamond\Box A \rightarrow \Diamond\Diamond\Box A$	1 par ($RM_\Diamond$)
	3.	$\Diamond\Diamond\Box A \rightarrow \Diamond\Box A$	($4_\Diamond$ appliqué à $\Box A$
	4.	$\Diamond\Box\Diamond\Box A \rightarrow \Diamond\Box A$	2 + 3

On a aussi la formule duale ($RS4_\Diamond$) $\Box\Diamond A \leftrightarrow \Box\Diamond\Box\Diamond A$. Donc :
- toutes les chaînes de modalités se réduisent à des chaînes alternées (par réduction de $\Box\Box$ et de $\Diamond\Diamond$)
- toutes les chaînes alternées de longueur 4 se réduisent à $\Box\Diamond$ ou $\Diamond\Box$ (les deux seules chaînes alternées de longueur 2)

donc ne sont distinctes que les chaînes alternées de longueur au plus 3 et leur négations. On a en plus des implications entre modalités qui se résument dans le schéma suivant :



Modalités de S4

Justification des flèches

$\Box\rightarrow$ $\Box\Diamond\Box$ cf (RS4)$\rightarrow$, ligne 4

$\Box\rightarrow$ $\bullet$ (T)

$\Box\Diamond\Box\rightarrow$ $\Diamond\Box$ (T) appliqué à $\Diamond\Box A$

$\Box\Diamond\Box\rightarrow$ $\Box\Diamond$ $RM_\Diamond$ puis RM appliqués à (T)

$\Diamond\Box\rightarrow$ $\Diamond\Box\Diamond$ même technique appliquée à ($T_\Diamond$)

$\Box\Diamond\rightarrow$ $\Diamond\Box\Diamond$ (T_{pos}) appliqué à $\Box\Diamond A$

$\Diamond\Box\Diamond\rightarrow$ $\Diamond$ dual de $\Box\rightarrow$ $\Box\Diamond\Box$

$\bullet\rightarrow$ $\Diamond$ ($T_\Diamond$)

8.3 Modalités des logiques incluant l'axiome (5)

8.3.1 Modalités de **K5**

On a donc les axiomes (K) et (5)$\Diamond A \rightarrow \Box\Diamond A$. Nous démontrons les 4 réductions ci-dessous :

1. $\Box\Box\Box A \leftrightarrow \Box\Box A$
2. $\Box\Diamond\Box A \leftrightarrow \Box\Box A$
3. $\Diamond\Box\Box A \leftrightarrow \Diamond\Box A$
4. $\Diamond\Diamond\Box A \leftrightarrow \Diamond\Box A$

Les 4 autres réductions des modalités de taille 3 se déduisent par dualité. Remarquez que toutes ces règles se résument à effacer les milieux.

2$\rightarrow$ et 4$\rightarrow$ résultent de ($5_\Diamond$) $\Diamond\Box A \rightarrow \Box A$ par RM et $RM_\Diamond$.

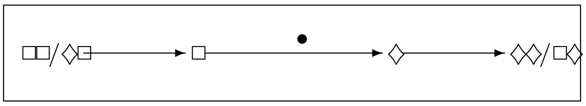
Les six autres implications utilisent $(\Diamond A \rightarrow \Box B) \rightarrow (\Box A \rightarrow \Box B)$ (a) et $(\Diamond A \rightarrow \Box B) \rightarrow (\Diamond A \rightarrow \Diamond B)$ (b) qui sont des théorèmes de K. La démonstration figure ci-dessous (Chellas 1980)

1	$\Diamond\Box A \to \Box A$	$5_\Diamond$
2	$\Box\Diamond\Box A \to \Box\Box A$	1, RM
3	$\Diamond\Diamond\Box A \to \Diamond\Box A$	1, $\mathrm{RM}_\Diamond$
4	$\Diamond\Box A \to \Box\Diamond\Box A$	(5) appliqué à $\Box A$
5	$\Diamond\Box\Box A \to \Box\Box A$	($5_\Diamond$) appliqué à $\Box A$
6	$\Diamond\Box A \to \Box\Box A$	2,4,LP
7	$\Box\Diamond\Box A \to \Box\Box\Box A$	6, RM
8	$\Diamond\Box A \to \Box\Box\Box A$	4,7,LP
9	$\Box\Box A \to \Box\Diamond\Box A$	4,(a), LP
10	$\Diamond\Box A \to \Diamond\Diamond\Box A$	4, (b), LP
11	$\Box\Box\Box A \to \Box\Box A$	5, (a), LP
12	$\Diamond\Box\Box A \to \Diamond\Box A$	5, (b), LP
13	$\Box\Box A \to \Box\Box\Box A$	8, (a), LP
14	$\Diamond\Box A \to \Diamond\Box\Box A$	8, (b), LP

Les modalités dans K5 sont représentées dans la figure 8.2

8.3.2 Modalités de **KD5**

On a en plus l'axiome (D) $\Box A \to \Diamond A$; donc en appliquant (D) à $\Box A$ on a $\Box\Box A \to \Diamond\Box A$, ce qui dans la figure 8.2 fusionne deux modalités. Par dualité, on a aussi $\Box\Diamond A \to \Diamond\Diamond A$. Compte tenu de (D), il reste :



8.3.3 Modalités de **K45**

On ajoute à **K5** l'axiome (4) $\Box A \to \Box\Box A$. La figure 8.2 se réduit en conséquence :

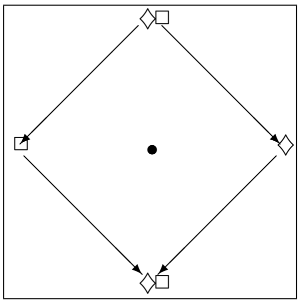
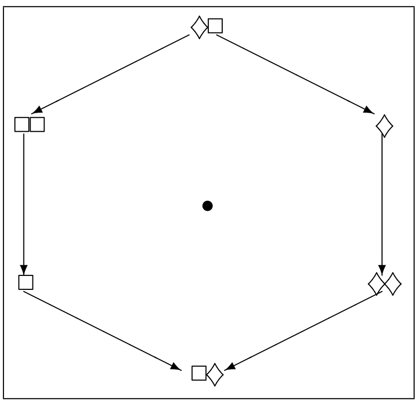
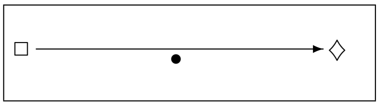


FIG. 8.2 – Modalités de K5



8.3.4 Modalités de **KD45**

On arrive au résultat en ajoutant (4) $\Box A \rightarrow \Box\Box A$ à $\mathsf{KD5}$. Il reste les même modalités que dans $\mathsf{S5}$, mais leur structure est différente – ce qui fait de $\mathsf{KD45}$ un des candidats quand $\mathsf{S5}$ ne convient pas et qu'on cherche la logique la plus simple possible. La structure est la suivante :



Cours 9

Logiques modales du temps

9.1 Intuitions

On ne peut pas considérer la représentation du temps comme *un* dommaine d'application : il y a peu de critères communs entre la modélisation de bon sens de la manipulation d'objets courants, celle de la sémantique du temps verbal, et celle d'un programme dont on veut montrer qu'il termine. On a besoin de représenter le temps, et divers formalismes ont été proposés pour cela qui ont des points de rencontre et qui diffusent plus ou moins d'une application à l'autre. La représentation du temps n'est pas non plus un problème purement formel : on verra que ce qu'il faut représenter n'est pas évident, et que même dans les domaines formels il y a plusieurs représentations du temps possibles. Enfin, dernier point, les logiques temporelles ne sont pas une sous-catégories des logiques modales. On peut traiter du temps au premier ordre, ou dans des formalismes dédiés dont les plus connus en IA sont le calcul des situatons [McCarthy], le calcul des événements [Kovalski et Sergot] et le calcul des fluents [Thielscher].

On passe facilement de l'intuition des mondes possibles à une représentation temporelle avec une idée simple : chaque monde possible est un état de chose à un instant donné, et la relation d'accessibilité représente l'ordre entre les instants. Ceci dit, ni la notion d'instant, ni les propriétés de l'ordre ne sont des évidences, et il y a des choix de représentation à faire. C'est ce point que nous allons d'abord explorer en expliquant quelques propriétés que peut avoir ou pas la notion de temps sous-jacente à une représentation, selon l'aspect que les auteurs auront privilégié.

9.1.1 temps linéaire vs temps branchu

Première intuition, le temps est une série de points de repère ordonnés. Mais sont ils tous comparables ? Une conception plus proche du temps physique répondra oui - le temps est une réalité objective commune à tous les objets. Une conception plus cognitive autorisera des espaces de repèrage indépendants les uns des autres ou seulement partiellement raccordés, qui se passent sur des plans différents.

Dès lors que les instants sont ordonnés et que la représentation reste formellement manipulable, on est contraint par les propriétés formelles des relations d'ordre. Rappellons en passant quelques définitions relatives à celles-ci :

- Une relation transitive est une relation de *préordre*. C'est la propriété essentielle qui mène à l'ordre. Si $<$ est un préordre, $((a<b \wedge b<a) \vee a=b)$ est une relation d"équivalence et la relation quotient de $(a<b \vee a=b)$ sur les classes d'équivalence est reflexive, symétrique et transitive : ce qu'on appelle un *ordre large*. Si $<=$ est un ordre large, la relation $a<b$ ssi $(a<=b \wedge \neg(a=b))$ est un *ordre strict* (antireflexif, transitif) ; inversement, si $<$ est un ordre strict, $a<=b$ ssi $(a<b \vee a=b)$ est un ordre large.
- Un ordre est *total* si tous les éléments distincts sont comparables deux à deux. Quand on trouve cette propriété trop contraignante, on peut ne l'imposer que pour des éléments comparables à un même troisième. Un ordre est *linéaire à gauche* si $(a<c \wedge b<c) \rightarrow (a<b \vee a=b \vee a>b)$. Symétriquement, il est *linéaire à droite* si $(c<a \wedge c<b) \rightarrow (a<b \vee a=b \vee a>b)$. Enfin, un ordre est *linéaire* si il est linéaire à gauche et à droite.
- Quelle est la différence entre un ordre linéaire et un ordre total ? Un ordre linéaire peut être composé de plusieurs ensembles dont chacun est totalement ordonné, mais qui sont sans rapport les uns aux

autres[1]. Un ordre est *connexe* si deux éléments quelconques sont toujours reliés par une chaîne de comparaisons : pour tout a, b, il existe $c_0 \ldots c_n$ tels que $c_i \bullet_i c_{i+1}$ avec c_0=a, c_n=b et $\bullet_i$ est alternativement < et >. Il est facile de montrer qu'un ordre total est un ordre linéaire connexe.

Quand on modélise le temps comme linéaire à gauche, un point a une seule chaîne d'instants passés, mais peut avoir plusieurs chaînes d'instants futurs non comparables entre eux. On parle alors souvent de *temps branchu vers le futur*. La notion de temps branchu vers le passé est symétrique mais moins usitée : il est plus courant de se placer dans une situation où le passé est connu et le futur indéterminé. Enfin, on parle en général de *temps linéaire* pour dire que tous les instants sont sur une seule ligne, donc que l'ordre est total !

9.1.2 temps discret, dense, complet

Seconde question sur la structure du temps, est-il atomique ou pas ? Traduit en termes plus modernes, peut-on numéroter les instants sur une branche dans leur ordre d'occurence ? C'est un second choix de représentation. Mathématiquement, il conduit à deux définitions. La première correspond au temps vu comme des tops d'horloge rhytmant par exemple l'exécution des instructions d'une machine, la seconde au temps qui mesure la quantité d'eau déversée par un robinet dont les conditions de fonctionnement ne varient pas.

L'ordre < est *discret* si $\forall$x,y x<y $\rightarrow$ $\exists$z,u (x<z$\leq$y $\wedge$ x$\leq$u<y $\wedge$ $\forall$t $\neg$(x<t $\wedge$ t<z) $\wedge$ $\forall$t $\neg$(u<t $\wedge$ t<y)). Il est *dense* si à l'inverse $\forall$x$\forall$y x<y $\rightarrow$ $\exists$z (x<z $\wedge$ z<y).

La caractéristique d'un ordre discret est que chaque élément qui n'est pas maximal a des successeurs et des predecesseurs immédiats (un seul si l'ordre est linéaire, éventuellement plusieurs sinon), alors que dans un ordre dense, il n'y a pas de successeur immédiat. La complétude vise un autre point, que l'on pourrait illustrer ainsi : si l'on a deux ensembles V et P d'instants tels que tout instant de V est avant tout instant de P (par exemple avant la lumière était éteinte et après elle est allumée), y a-t-il toujours au moins un instant qui les sépare ? Pour en rester aux nombres, on sait que cette propriété est vérifiée dans les réels ($\Re$) mais pas dans les rationnels ($\mathcal{Q}$) : dans le premier ensemble, il y a un nombre qui sépare ceux dont le carré est supérieur à 2 et ceux dont le carré est inférieur à 2 ; dans le second ensemble, ce n'est pas le cas.

9.1.3 Points vs intervalles

Dernier choix de représentation : on peut définir un intervalle comme tout ce qui est entre deux points, ou bien définir un point comme la borne d'un intervalle. Mais quelle est la notion primitive ? Les deux choix sont possibles : dans la première conception, la sémantique du temps sera un ensemble de points muvis d'une relation < transitive ; dans la seconde, c'est un ensemble d'intervalles muni de plusieurs relations. Le cas le plus typique est l'ensemble des relations d'Allen : p (précedes), m (meet), o (overlaps), s (starts), d (during), = (equals), e (ends) et leurs inverses.

9.1.4 Et la durée ?

9.2 Les logiques temporelles de base

Comment traduire en logique modale les relatons temporelles ? On va considérer que chaque instant de référence (point ou intervalle) est un monde, et la relation d'ordre entre instants se reflète dans l'accessibilité. Il y a quand même un problème : x<y peut être vu du point de vue de x ou de celui de y : y est dans le futur de x ou x est dans le passé de y.

Il faut donc deux modalités réciproques, que l'on appelle traditionnellement P et F. Ce sont des modalités possibles : $P\mathrm{p}_0$ signifie p_0 est valide dans un monde passé, $F\mathrm{p}_0$ signifie p_0 est valide dans un monde futur. Leur relation d'accessibilité est définie pour P par $\mathcal{R}(w, w')$ si w'<w, pour F par $\mathcal{R}(w, w')$ si w<w'.

On définit les modalités duales qui sont nécessaires : $H\mathrm{p} \stackrel{def}{=} \neg P \neg \mathrm{p}$ (H pour 'has always been') et $G\mathrm{p} \stackrel{def}{=} \neg F \neg \mathrm{p}$ (G pour 'going always to be'). La logique s'appelle $\mathsf{K_t}$ (la logique de Prior). Elle s'axiomatise en ajoutant aux axiomes habituels des logiques standard la relation de réciprocité entre le passé et le futur qui établit qu'un monde est dans le passé de tous ses futurs et dans le futur de tous ses passés :

[1]Malheureusement, les anglais utilisent couramment ordre linéaire pour ordre strict !

$H(\mathrm{A} \rightarrow \mathrm{B}) \rightarrow (H\mathrm{A} \rightarrow H\mathrm{B})$ $\qquad$ $G(\mathrm{A} \rightarrow \mathrm{B}) \rightarrow (G\mathrm{A} \rightarrow G\mathrm{B})$

$\dfrac{A}{HA}$ $\qquad$ $\dfrac{A}{GA}$

$\mathrm{A} \rightarrow HF\mathrm{A}$ $\qquad$ $\mathrm{A} \rightarrow GP\mathrm{A}$

Notez que $\mathsf{K_t}$ est correct et complet pour la classe des cadres qui sont des relations irreflexives i.e. $\neg\mathcal{R}(w,w)$, mais qu'aucune shchéma d'axiome ne peut caractériser cette classe de cadre. A ce point, on n'a pas véritablement une logique temporelle, puisqu'on n'a que la réciprocité. Nous donnons ci-dessous différents axiomes qui peuvent compléter la logique de Prior :

Propriété	Axiome
Réflexivité	$H\mathrm{A} \rightarrow \mathrm{A}$ ou $G\mathrm{A} \rightarrow \mathrm{A}$ (T)
Transitivité	$H\mathrm{A} \rightarrow HH\mathrm{A}$ ou $G\mathrm{A} \rightarrow GG\mathrm{A}$ (4)
Linéarité à droite	$(F\mathrm{A} \wedge F\mathrm{B}) \rightarrow (F(\mathrm{A} \wedge F\mathrm{B}) \vee F(\mathrm{A} \wedge \mathrm{B}) \vee F(F\mathrm{A} \wedge \mathrm{B}))$ $(.3_d)$
Linéarité à gauche	$(P\mathrm{A} \wedge P\mathrm{B}) \rightarrow (P(\mathrm{A} \wedge P\mathrm{B}) \vee P(\mathrm{A} \wedge \mathrm{B}) \vee P(P\mathrm{A} \wedge \mathrm{B}))$ $(.3_g)$
densité	$P\mathrm{A} \rightarrow PP\mathrm{A}$ ou $F\mathrm{A} \rightarrow FF\mathrm{A}$ (den)
Pas d'instant maximal	$G\mathrm{A} \rightarrow F\mathrm{A}$ (D_d)
Pas d'instant minimal	$H\mathrm{A} \rightarrow P\mathrm{A}$ (D_g)

Le minimum pour avoir une notion de flux temporel qui soit un ordre est d'ajouter la transitivité. La plus petite logique qui mérite ce nom est donc $\mathsf{K_t4}$. Notez qu'on ne peut pas caractériser l'ordre total : il n'y a aucune formule F telle que, si $\mathcal{M} \models$F, alors F est un ordre total.

Une autre logique intéressante est $\mathsf{K_tQ}$ dont les axiomes sont ceux de $\mathsf{K_t}$ plus (4), (D_d), (D_g), (den), $(.3_d)$ et $(.3_g)$ qui caractérise un ordre linéaire dense sans extrémum.

9.3 Since et until

9.3.1 De nouveaux connecteurs

Les opérateurs P et F ne représentent qu'une part des relations temporelles intéressantes. Nous introduisons ici du point de vue sémantique Until, Since et Next. Les définitions ci dessous suffisent quand le temps est linéaire :

U(A,B) valide en w *ssi* il existe w'>w t.q. A valide en w' et pour tout u t.q. w<u<w', B valide en u

S(A,B) valide en w *ssi* il existe w'<w t.q. A valide en w' et pour tout u t.q. w'<u<w, B valide en u

NA valide en w *ssi* il existe w'>w t.q. A valide en w' et $\{u|w' > u > w\} = \emptyset$

S et U deviennent les modalités fondamentales : on peut définir P , F et N en termes de S et U : PA $\leftrightarrow S(\mathrm{A},\top)$ et FA $\leftrightarrow U(\mathrm{A},\top)$. NA est sémantiquement équivalent à $U(\mathrm{A},\perp)$. Par contre, on ne peut définir Since, Until ou Next avec une formule finie utilisant uniquement P et F . On ne peut pas non plus définir Until avec une formule utilisant uniquement P , F et N.

Quand le temps est branchu (vers le passé ou vers le futur), il y a deux idées distinctes qu'on peut chercher à représenter avec ces connecteurs : soit un au moins des futurs (passés) vérifie la propriété, soit elle est valide dans *tous* les futurs (passés). La définition ci-dessus traduit la première idée. Si l'on pense à toutes les branches, la définition devient, en écartant comme dans le cas totalement ordonné la validité dans les mondes aveugles :

U'(A,B) valide en w ssi	a) il y a au moins un v t.q. v>w, b) pour tout v>w, il existe w'>w t.q. w' est comparable à v (v>w'$\vee$ v=w' $\vee$ v<w'), A est valide en w' et pour tout u t.q. w<u<w', B est valide en u
S'(A,B) valide en w ssi	a) il y a au moins un v t.q. v>w, et b) pour tout v<w, il existe w'<w t.q. w' est comparable à v (v>w'$\vee$ v=w' $\vee$ v<w'), A est valide en w' et pour tout u t.q. w'<u<w, B est valide en u
N'A valide en w ssi	a) il y a au moins un v t.q. v>w, et b) pour tout v>w, il existe w' t.q. v$\geq$w'>w, A valide en w' et $\{u|w' > u > w\} = \emptyset$

Voici quelques axiomes qui décrivent les relations entre ces connecteurs modaux pour un temps linéaire :

$G(\mathrm{A} \to \mathrm{B}) \to (U(\mathrm{A,C}) \to U(\mathrm{B,C}))$	$\mathrm{A1}_f$
$G(\mathrm{A} \to \mathrm{B}) \to (U(\mathrm{C,A}) \to U(\mathrm{C,B}))$	$\mathrm{A2}_f$
$\mathrm{A} \wedge U(\mathrm{B,C}) \to U(\mathrm{B} \wedge S(\mathrm{A,C}),\mathrm{C})$	$\mathrm{A3}_f$
$U(\mathrm{A,B}) \wedge \neg U(\mathrm{A,C}) \to U(\mathrm{B} \wedge \neg \mathrm{C}, \mathrm{B})$	$\mathrm{A4}_f$
$U(\mathrm{A,B}) \to U(\mathrm{A}, \mathrm{B} \wedge U(\mathrm{A,B}))$	$\mathrm{A5}_f$
$U(\mathrm{B} \wedge U(\mathrm{A,B}),\mathrm{B}) \to U(\mathrm{A,B})$	$\mathrm{A6}_f$
$U(\mathrm{A,B}) \wedge U(\mathrm{C,D}) \to U(\mathrm{A}\wedge \mathrm{C},\mathrm{B}\wedge \mathrm{D}) \vee U(\mathrm{A}\wedge \mathrm{D},\mathrm{B} \wedge \mathrm{D}) \vee U(\mathrm{B} \wedge \mathrm{C},\mathrm{B} \wedge \mathrm{D})$	$\mathrm{A7}_f$

9.3.2 La logique de Manna et Pnueli (1981)

C'est une logique pour la vérification de systèmes concurents à variables partagées. On travaille donc sur du temps entier – on verra tout à l'heure comment construire un modèle. Les opérateurs modaux sont $\Box$A (A sera toujours vrai <u>y compris maintenant</u>), $\Diamond$A (A sera parfois vrai, <u>y compris maintenant</u>), NA (A est vrai au prochain instant), U^*(A,B) (until : B est vrai jusqu'à ce que A soit vrai).

Les axiomes de cette logique sont :

$$
\begin{array}{l}
\neg\Diamond \mathrm{A} \ \leftrightarrow \Box\neg \mathrm{a} \\
\Box(\mathrm{A} \to \mathrm{B}) \to (\Box \mathrm{A} \text{ imp } \Box \mathrm{B}) \\
\Box \mathrm{A} \to \mathrm{A} \\
N\neg \mathrm{A} \ \leftrightarrow \neg N\mathrm{A} \\
N(\mathrm{A} \to \mathrm{B}) \to (N\mathrm{A} \to N\mathrm{B}) \\
\Box \mathrm{A} \to N\mathrm{A} \\
\Box \mathrm{A} \to N\Box \mathrm{A} \\
(\mathrm{A} \wedge \Box(\mathrm{A} \to N\mathrm{A})) \to \Box \mathrm{A} \\
U^*(\mathrm{A,B}) \ \leftrightarrow \mathrm{A} \vee (\mathrm{B} \wedge NU^*(\mathrm{A,B})) \\
U^*(\mathrm{A,B}) \to \Diamond \mathrm{A}
\end{array}
$$

Les règles d'inférence sont standard : modus ponens et nécessitation.

On laisse comme un exercice la justification de chacun des axiomes par rapport à une représentation du temps. Nous donnons seulement une sémantique pour cette logique. Un monde (donc un instant) est une suite infinie d'entiers $\mathrm{s}=(\mathrm{x}_0,\mathrm{x}_1,\mathrm{x}_2,\dots)$. On note s^n l'instant $\mathrm{s}^\mathrm{n}=(\mathrm{x}_\mathrm{n},\mathrm{x}_{\mathrm{n}+1},\dots)$, autrement dit la queue de s à partir de x_n. La définition de la validité des atomes et des connecteurs propositionnels est la même que d'habitude. Voici celle de la validité des connecteurs modaux :

$\mathcal{M} \models_s \Box \mathrm{A}$	ssi	pour tout n, A est valide en s^n
$\mathcal{M} \models_s \Diamond \mathrm{A}$	ssi	il y a un moins un n t.q. A est valide en s^n
$\mathcal{M} \models_s N\mathrm{A}$	ssi	A est valide en s_1
$\mathcal{M} \models_s U^*(\mathrm{A,B})$	ssi	il y a un n t.q. A soit valide en s^n et pour tous les m t.q. m<n, B est valide en s^m

Cours 10

Logiques modales de l'action

10.1 Logique propositionnelle dynamique (PDL)

PDL est au départ une logique des programmes, qui peut se voir aussi comme une logique modale de l'action. Au lieu que la modalité soit déclenchée par le top d'horloge comme dans la logique de Manna et Pnueli, où l'on pense plutôt à l'analyse des traces sans savoir quelle instruction a été exécutée, dans PDL on raisonne au niveau modal sur les instructions et leur combinaison dans des programmes. On s'est rendu compte ensuite que, si l'on regarde de près, on raisonne sur des combinaisons de transformations de l'état du monde – mais rien dans le formalisme n'impose que ces transformations soient l'oeuvre d'un programme, dès lors que l'état du monde s'exprime en LP. Une seconde interprétation de PDL intéressante pour nous est de considérer qu'il s'agit de combinaisons d'actions (des plans).

La première version à analyser est la version standard de PDL ("regular PDL"). On se donne un ensemble des propositions atomiques $\mathcal{P} = \{p_j\}_{j\in J}$ et un ensemble de programmes atomiques (instructions / actions) $\mathcal{A} = \{a_i\}_{i\in I}$. Les instructions sont *a priori* non déterministes. Les fbf de PDL sont formées à partir de constructeurs de formules et de constructeurs de programmes par les règles récursives suivantes :

$$\begin{array}{l} \pi = a_i \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \\ \mathrm{F} = \mathrm{p}_j \mid \neg\mathrm{F} \mid \mathrm{F}_1 \wedge \mathrm{F}_2 \mid \mathrm{F}_1 \vee \mathrm{F}_2 \mid \mathrm{F}_1 \rightarrow \mathrm{F}_2 \mid \mathrm{F}_1 \leftrightarrow \mathrm{F}_2 \mid [\pi]\mathrm{F} \mid \langle\pi\rangle\mathrm{F} \end{array}$$

; est l'exécution séquentielle : $\pi_1;\pi_2$ est le programme (ou plan) formé de π_1 suivi de π_2. $\cup$ est le choix (on exécute l'un des deux), * est l'itération : π^* est le programme / plan qui répète π un nombre indéterminé de fois (ce peut être 0). Notez que la première ligne définit à elle seule l'ensemble Π des programmes (ou plans). $[\pi]$A est une formule qui est vraie si A est vraie après toutes les exécutions du programme π, $\langle\pi\rangle$A est vraie si A est vraie après au moins une exécution du programme π.

Par rapport aux logiques modales "de base", nous avons maintenant une infinité de modalités, et il faut donc redéfinir la notion de modèle pour l'adapter à cette extension. On aura pour cela deux catégories de relations d'accessibilité : celles qui correspondent aux instructions atomiques, et celles qui correspondent aux plans composés.

Un modèle $\mathcal{M} = \langle \mathcal{W}, \{\mathcal{R}_\pi | \pi \in \Pi\}, \mathcal{V}\rangle$ comporte les mêmes ensembles de mondes et de valuation des atomes propositionnels qu'un modèle standard. Les relations d'accessibilité pour les programmes non atomiques doivent satisfaire les contraintes suivantes :

$$\begin{aligned} \mathcal{R}_{\pi_1\cup\pi_2} &= \mathcal{R}_{\pi_1} \cup \mathcal{R}_{\pi_2} \\ \mathcal{R}_{\pi_1;\pi_2} &= \mathcal{R}_{\pi_1} \circ \mathcal{R}_{\pi_2} \text{ (composée)} \\ \mathcal{R}_{\pi^*} &= (\mathcal{R}_\pi)^* \text{ (cloture reflexive et transitive)} \end{aligned}$$

Pour lire ces contraintes, pensez qu'une relation est un ensemble de couples de mondes. O : on définit la relation composée par $\mathcal{R}_1 \circ \mathcal{R}_2 = \{(w,w')|\exists u(\mathcal{R}_1(w,u) \wedge \mathcal{R}_2(u,w'))$ et la cloture reflexive et transitive par $(\mathcal{R})^* = \bigcup_{i\geq 0} \mathcal{R}^i$, où $\mathcal{R}^i$ est la composée i fois de $\mathcal{R}$ et $\mathcal{R}^0$ est la relation diagonale.

La définition de la satisfaction dans ce modèle répète pour chaque programme la définition habituelle des modèles standard. Si on a $R_\pi(w,w')$ et $\mathcal{M} \models_{w'}$ A, on a $\mathcal{M} \models_w \langle\pi\rangle$A. Si aucun monde n'est accessible depuis w par R_π, on dit que le programme π échoue en w ou n'est pas exécutable en w. Notez que dans ce cas on a pour tout A $\mathcal{M} \models_w [\pi]$A et $\mathcal{M} \not\models_w \langle\pi\rangle$A.

Notez aussi tout de suite que l'ajout de la cloture reflexive et transitive rend la détection de contradictions plus difficile, parce qu'un ensemble infini de formules peut être inconsistant sans qu'aucun de ses sous-ensembles finis le soit. Par exemple, $\Gamma = \{\langle a^*\rangle B, \neg B, \neg\langle a\rangle B, \neg\langle a\rangle\langle a\rangle B, \neg\langle a\rangle\langle a\rangle\langle a\rangle B, \ldots\}$ est inconsistant, mais tous ses sous-ensembles finis sont satisfiables.

La logique PDL (du moins la version standard) est axiomatisable. Les axiomes de PDL sont :

$$
\begin{array}{lr}
[\pi](\mathrm{A} \to \mathrm{B}) \to ([\pi]\mathrm{A} \to [\pi]\mathrm{B}) & \text{(i)} \\
\langle\pi\rangle\mathrm{A} \ \leftrightarrow \neg[\pi]\neg\mathrm{A} & \text{(ii)} \\
\langle\pi_1;\pi_2\rangle\mathrm{A} \ \leftrightarrow \langle\pi_1\rangle\langle\pi_2\rangle\mathrm{A} & \text{(iii)} \\
\langle\pi_1 \cup \pi_2\rangle\mathrm{A} \ \leftrightarrow \langle\pi_1\rangle\mathrm{A} \vee \langle\pi_2\rangle\mathrm{A} & \text{(iv)} \\
\langle\pi^*\rangle\mathrm{A} \ \leftrightarrow (\mathrm{A} \vee \langle\pi\rangle\langle\pi^*\rangle\mathrm{A}) & \text{(v)} \\
[\pi^*](\mathrm{A} \to [\pi]\mathrm{A}) \to (\mathrm{A} \to [\pi^*]\mathrm{A}) & \text{(vi)}
\end{array}
$$

L'axiome (vi) est l'axiome d'induction de Segerberg. Avec le (v), il suffit à caractériser les cadres réguliers, i.e. ceux où π^* est interprété par la cloture reflexive transitive de π.

Comme on reste dans le cadre des logiques normales (celles qui décrivent les modèles standard), et qu'on n'a pas imposé de contrainte sur les relations $[\pi]$, seul l'axiome (K) s'impose à chaque modalité : c'est (i). (ii) est un axiome nouveau que vouus commenterez en exercice. Les autres axiomes décrivent les constructeurs de programmes.

Les régles de déduction seront celles de la logique K. $\pi_1;\pi_2$ et $\pi_1 \cup \pi_2$ s'y réduisent par les axiomes (iii) et (iv). π^* pose par contre des problèmes plus difficiles ...

10.2 Variantes de PDL

PDL désigne en pratique toute une famille de logiques, parce qu'on peut considérer d'autres constructeurs de programmes. En voici trois :

Intersection Le programme est noté $\pi_1 \cap \pi_2$, avec la définition sémantique naturelle $R_{\pi_1 \cap \pi_2} = R_{\pi_1} \cap R_{\pi_2}$. Autrement dit, deux états sont reliés par le programme intersection ssi ils sont reliés par chacun des programmes. On explique souvent cette relation en disant qu'il s'agit d'une exécution parallèle. Notons que ce n'est correct que dans une sémantique des traces - on a pu passer de tel à tel état par l'un ou l'autre programme.

Négation On note $\neg\pi$ le programme dont la relation est le complémentaire de celle de π : $R_{\neg\pi} = \mathcal{W} \times \mathcal{W} - R_\pi$. Ce constructeur est employé pour formaliser des questions de planification où l'on veut disposer d'une commande *ne pas faire* π. Même en limitant π à des ensembles d'actions atomiques, il pose des problèmes à la fois de calcul et d'interprétation qui en limitent l'usage.

Test La version standard de PDL n'a pas la possibilité de décrire une conditionnelle. Il faut introduire pour cela un type d'action particulier : des actions qui ne changent rien à l'état du monde, mais produisent seulement de l'information. Pour cela, on introduit pour chaque formule A une action A ? qui réussit si A est vrai dans le monde actuel et qui échoue sinon. On définit donc $R_{A?} \stackrel{def}{=} \{(w,w) | \mathcal{M} \models_w A\}$.

Le test est le constructeur le plus important. Son utilisation n'est pas évidente : si l'on écrit une formule $\langle A?\rangle$B, elle équivaut à A $\wedge$ B, alors que $[A?]$B équivaut à A $\to$ B. L'utilité de cette modalité vient de ce qu'elle filtre tous les chemins d'exécution où elle échoue : l'instruction suivante n'est pas atteinte sur ce chemin. Om modélise la construction :

```
si A
faire π1
sinon
faire π2
fin si
```

par le programme (?A ;π_1)$\cup$(?$\neg$A ;π_2).

Cours 11

Logiques non modales de l'action

Ce point ne peut qu'être effleuré. Nous voudrions surtout mettre en évidence l'intition commune sous-jacente à des formalismes différents : l'action fait passer d'un état du monde à un autre. Ceci étant, les formalismes développés en I.A. s'attachent à un point que la logique modale ne traite pas assez bien pour les besoins de l'I.A. : la description précise de ce qui change et ce qui ne change pas lors d'une action.

Pour comprendre ce nouveau formalisme, nous passerons par une étude purement logique : peut-on trouver à la logique modale un équivalent en logique du 1er ordre.

11.1 Traduction de la logique modale en LPO

La question de la traduction se décompose en deux :

- peut-on donner une correspondance systématique entre les formules modales et *des* formules de LPO ?
- cette correspondance préserve-t-elle les propriétés de déduction, i.e. les conséquences des formules traduites sont-elles les traductions des conséquences dans le formalisme originel ?

Dans un premier temps, on donne une transformation des formules de logique modale en formules de LPO. L'idée est qu'à chaque atome propositionnel p_i correspond un prédicat $P_i()$ d'arité 1. On pourra ainsi étudier une équivalence sémantique entre p_i et $P_i(x)$ en faisant varier la variable x dans $\mathcal{W}$et en interprétant $P_i()$ par l'ensemble des mondes w t.q. $\mathcal{M} \models_w p_i$.

Nous définissons donc d'abord la transformation, appelée *traduction standart*. Bien entendu, c'est une définition récursive par rapport à la structure de la formule. A cela s'ajoute deux petites subtilités : la traduction utilise un unique symbole de relation R, le même dans chacune des règles de la définition ; elle utilise aussi dans chaque règle une variable libre que l'on peut nommer comme on veut, à condition de ne pas créer de collision avec les variables déja utilisées. On notera $\mathrm{Trad}_x(\mathrm{F})$ la traduction de la formule F utilisant la variable x

Formute modale F	**traduction $\mathrm{Trad}_x(\mathrm{F})$**
p_i	$P_i(x)$
$\perp$	$x \neq x$
$\neg A$	$\neg \mathrm{Trad}_x(A)$
$A \vee B$	$\mathrm{Trad}_x(A) \vee \mathrm{Trad}_x(B)$
$\Diamond A$	$\exists y\ (R(x,y) \wedge \mathrm{Trad}_y(A))$

On n'a défini la traduction que pour les connecteurs logiques $\neg$ et $\vee$ et l'opérateur $\Diamond$, puisque les autres peuvent être considérés comme des abréviations. On démontrerait sans aucun problème que $\mathrm{Trad}_x(A \wedge B) = \mathrm{Trad}_x(A) \wedge \mathrm{Trad}_x(B)$ ou que $\mathrm{Trad}_x(A \rightarrow B) = \mathrm{Trad}_x(A) \rightarrow \mathrm{Trad}_x(B)$. De même, en utilisant la définition de $\Box$à l'aide de $\Diamond$, on obtient facilement $\mathrm{Trad}_x(\Box A) = \forall y\ (R(x,y) \rightarrow \mathrm{Trad}_y(A))$.

Si l'on traduit par exemple les deux axiomes (T) et (4) appliqués aux propositions atomiques p et q, on obtient :

1. Cas de (T) :

Formute modale F	traduction $\mathbf{Trad}_x\mathbf{(F)}$
p → q □p □q	P(x) → Q(x) ∀y (R(x,y) → P(y)) ∀y (R(x,y) → Q(y))
□(p → q)	∀y (R(x,y) → (P(y) → Q(y)))
□p → □q	(∀y (R(x,y) → P(y))) → (∀y (R(x,y) → Q(y)))
T(p,q)	∀y (R(x,y) → (P(y) → Q(y))) → (∀y (R(x,y) → P(y))) → (∀y (R(x,y) → Q(y)))

2. Cas de (4)

Formute modale F	traduction $\mathbf{Trad}_x\mathbf{(F)}$
□p	∀y (R(x,y) → P(y))
□□p	∀y (R(x,y) → (∀z (R(y,z) → P(z)))) ∀y ∀z (R(x,y) → (R(y,z) → P(z)))
4(p)	∀y (R(x,y) → P(y)) → ∀y ∀z (R(x,y) → (R(y,z) → P(z)))

On notera que la traduction de (T) est une tautologie, mais pas celle de (4). En fait, on a un résultat d'équivalence : soit $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$ un modèle modal ; on peut interpréter le langage de la traduction au 1er ordre sur un modèle $\mathcal{M}_1$ d'univers $\mathcal{W}$ en faisant varier les variables x, y, z... dans $\mathcal{W}$ et en interprétant R par $\mathcal{R}$ et P_i par $\mathcal{V}(p_i)$. On a alors pour toute formule F : $\mathcal{M} \models_w$ F ssi $\mathcal{M}_1 \models \mathrm{Trad}_w(F)$ et $\mathcal{M} \models$ F ssi $\mathcal{M}_1 \models \forall x\ \mathrm{Trad}_x(F)$.

La question est alors : quelle est la portée de cette équivalence ? Toutes les formules de LPO dans le langage avec R et les P_i ne sont pas la traduction des formules modales : par exemple $\forall x\ P_0(x)$ ou ∀x ∀y R(x,y). Nous ne poursuivrons pas l'étude du sous-ensemble de ces formules qui sont la traduction d'une formule modale ; en l'état, elle montre l'analogie profonde entre la notion de monde possible et celle de situation que nous allons aborder maintenant – et aussi les différences qui portent principalement sur les formes d'inférence.

11.2 Calcul des situations

11.2.1 Formalisme de description

Le formalisme descriptif du calcul des situations est facile à comprendre. On veut décrire des actions, et comme en PDL, les actions sont des relations entre états du monde appelés maintenant *situations*[1] Dans chaque situation, on veut disposer d'un langage du premier ordre pour paramètrer actions et propriétés, pour donner un équivalent par exemple de $\models_s$ Rouge(voitureA), ou pour écrire que on obtient s' quand on est dans s et que la voiture B change de file.

Ajouter la situation en paramètre comme dans la traduction standard aurait été possible, mais en ferait un paramètre comme les autres. Nous adoptons la solution de M. Shanahan, qui est maintenant largement répandue et facilite les raisonnements. Elle consiste à passer au second ordre : on utilise une relation binaire Holds(propriété, situation) dont le premier argument est une relation sans argument de situation (par exemple Rouge(voitureA)). Par contre, dans la version de base les actions sont déterministes et donc la relation d'accessibilité est fonctionnelle : il y a une seule situation qui est le résultat de l'action a effectuée dans la situation s. Pour préserver l'indépendance entre la description de l'action et la situation, on utilise une fonction Result(action, situation) dont l'image est une situation.
On décrira par exemple une situation s_0 par :

Holds(Rouge(voitureA), s_0)
Holds(Lieu(voitureA, nationale 7), s_0)
Holds(Position(voitureA, pos_0), s_0)
Holds(Vitesse(voitureA, 50), s_0)
Holds(Distance(pos_0, carrefour, 300m), s_0)
etc.

et une action par :

Holds(Vitesse(voitureA, 0), Result(Stoppe(voitureA), s))

[1]Le calcul des situations remonte à J. Mc Carthy et P. Hayes entre 1963 et 1980 pour l'essentiel. Il ne faut pas le confondre avec la théorie des situations (*situation theory*) de J. Barrwise et J. Perry, qui utilise aussi le terme de 'situaton' dans un formalisme dont les préoccupations sont autres.

Holds(Vitesse(voitureA, v), s) → Holds(Vitesse(voitureA, v-30), Result(Freine(voitureA), s))
etc.

11.2.2 Déduction

11.2.2.1 Déduction classique et problème du cadre

La déduction classique est toujours valide dans cette théorie. On aura de façon élémentaire Holds(Vitesse(voitureA, 20), Result(Freine(voitureA), s_0)), i.e. après avoir freiné en s_0, la voiture A roule à 30 dans la situation résultante. La difficulté vient de ce qui a changé en même temps.

La distance de la voiture A au carrefour a du diminuer - que l'exemple ne dise pas dans quelle mesure est une simplification que l'on pourrait en partie lever, bien que le temps discret ne soit pas compatible avec une vision fine de cet aspect. Mais avant cela, le fait que la voiture soit rouge et qu'elle soit sur la nationale 7 ne sont pas affectés par le coup de frein. Or préciser cela serait très couteux en espace (il faut dire pour chaque action tous les prédicats qui ne sont pas modifiés) et très peu modulaire (chaque fois que l'on a besoin d'un nouveau prédicat, par exemple APotCatalytique(voitureX), il faut reformuler les axiomes en précisant s'il est affecté par chaque action, par exemple freiner.

Le problème du cadre est une des motivations qui ont poussé l'IA à travailler sur les logiques non monotones, i.e. à tenter de déplacer le problème (et sa solution) vers les règles de déduction en autorisant à déduire des conséquences du fait qu'aucun effet n'est décrit. Bien entendu, si on modifie la description en ajoutant de *nouveaux* effets, les conséquences précédentes devront être *rétractées* – d'où le terme de logique non-monotone.

11.2.2.2 La circonsription

La circonscription est un des systèmes d'inférence proposé pour avoir un calcul où rien ne change en dehors des règles. modifications explicites. L'idée est de minimiser l'extension de certains prédicats. On fixe deux ensemble de prédicats $\mu = \{P_1, \ldots, P_n\}$, $\nu = \{Q_1, \ldots, Q_m\}$ (μ = prédicats à minimiser, ν = prédicats vatiables) qui définissent un ordre $\leq_{\mu,\nu}$ sur les interprétations : $I_1 \leq_{\mu,\nu} I_2$ ssi ∀P ∈ μ, $I_1(P) \subseteq I_2(P)$ et ∀P' $\notin (\mu \cup \nu)$, $I_1(P') = I_2(P')$. La circonsription de F par rapport à μ, ν étant autorisé à varier, est la théorie des modèles de F minimaux pour $<_{\mu,\nu}$.

11.2.2.3 Conséquences et paradoxes

L'idée est que *normalement*, une propriété donnée n'est pas modifiée par l'action qui se déroule dans la situation donnée. On va pour cela introduire un prédicat Ab(action, propriété, situation) (Ab pour 'anormal'). On ajoute le schéma d'axiome

$$\neg Ab(a, P, s) \rightarrow \ [Holds(P, Result(a, s)) \leftrightarrow \ Holds(P, s)]$$

et on raisonne en minimisant le prédicat ab() et en ne laissant varier que Holds(). Ainsi, on minimise les cas anormaux qui sont les seuls où un prédicat peut évoluer. Dans l'exemple, la vitesse de la voiture peut diminuer, mais ni sa couleur, ni la route sur laquelle elle est ne peuvent changer – ce que l'on souhaitait.

Tous les problèmes ne sont pas pour autant résolus. Prennons le simple exemple suivant : en s_0, la voiture A est à l'arrêt au point mort. On a deux règles : passer la première met en prise, embrayer quand on est en prise met la voiture en mouvement. Que se passe-t-il si l'on passe la première et que l'on embraye ? Une des solutions est que la voiture est en prise en s_1 et en mouvement en s_2. Une autre solution, elle aussi minimale, est que la voiture est en prise en s_1 et est spontanément revenue au point mort en s_2 ...