Les bases de la logique modale

illustrées par des applications
en intelligence artificielle
et en génie logiciel

Davide CATTA Christian RETORÉ

Écrit entre le 20 Décembre 2021 et le 1er mai 2023

Table des matières

Chapitre 1

Modalités et logique

1.1 Modalités

Quelques exemples de modalités :

Logique	symbole	Lecture intuitive
Logique modale		
	$\Box$	il est nécessaire que
	$\Diamond$	il est possible que
Logique temporelle de Prior		
	G	il sera toujours vrai que
	F	il sera vrai que
	H	il a toujours été vrai que
	P	il a été vrai que
Logique épistémique		
	K_a	l'agent a sait que
	$\Box[i]$	transitions

Une modalité est un connecteur unaire, et ceux-ci vont par paires : car $\neg ModA \equiv DualMod\neg A$ exemple avec Carré et Losange

Un des guides pour construire une logique modale, est que les modalités doivent venir par paires :

$$\neg\Box A \equiv \Diamond \neg A$$

$$\neg\Diamond A \equiv \Box \neg A$$

et satisfaire le carré des oppositions.

La vérité d'une formule dans une situation (valuation) dépend de la vérité d'autres formules dans autres situations (valuations).

1.2 Le langage modal

Les formules logiques sont construites à partir d'un ensemble $\mathcal{P}$ au plus dénombrable d'atomes (on dit aussi variables propositionnelles ou lettres propositionnelles).

Dans la suite, nous considérons le connecteur unaire $\neg$ (négation), les connecteurs binaires $\wedge$ (conjonction), $\vee$ (disjonction) et $\rightarrow$ (implication ou conditionnel), ainsi que deux modalités $\Diamond$ (possible) et $\Box$ (nécessaire)

Solution: hello

Première partie

Logiques modales normales

Chapitre 2

Modèles de Kripke

2.1 Introduction

Différents moyens permettent de spécifier la signification des formules modales. On peut néanmoins dire que les modèles de Kripke se sont imposés comme l'approche dominante à la sémantique des formules modales sans doute à cause de la simplicité et élégance des définitions de base, qui, de plus, permet aisément de dériver des théorèmes de complétude.

2.2 Modèles de Kripke

Afin de définir l'interprétation d'une formule modale, nous supposerons donné un langage modal $\mathcal{L}$ défini à partir d'un ensemble de propositions $\mathcal{P}$, du connecteur unaire négation $\neg$, des connecteurs logiques binaires $\wedge, \vee, \rightarrow$ et des modalités $\Box, \Diamond$.

La première notion nécessaire est celle de cadre.

Définition 1 (Cadre)**.** *Un* cadre $\langle \mathcal{M}, \mathrm{R} \rangle$ *est constitué*

- *d'un ensemble non vide* $\mathcal{M} = \{\mathfrak{m}_i | i \in I\}$ *dont les membres sont généralement appelés mondes possibles ou situations,*
- *d'une relation binaire* R *sur* $\mathcal{M}$*, c'est-à-dire un sous ensemble* R *de* $\mathcal{M} \times \mathcal{M}$ *appelée accessibilité.*

On notera que R est une relation binaire quelconque. Il est tout à fait possible que $\mathfrak{m}_i \,\mathrm{R}\, \mathfrak{m}_i$, que R ait des points sans successeur, des points sans prédécesseur, etc.

Lorsque $\mathfrak{m}_i \,\mathrm{R}\, \mathfrak{m}_j$ nous dirons que le monde $\mathfrak{m}_j$ est accessible depuis le monde $\mathfrak{m}_i$.

Définition 2. *[Modèle de Kripke] Étant donné un cadre* $\langle \mathcal{M}, \mathrm{R} \rangle$ *un modèle de Kripke de cadre* $\langle \mathcal{M}, \mathrm{R} \rangle$ *s'obtient par la donnée d'une relation binaire* $\Vdash_0$ *appelée* forcing atomique*, entre mondes possibles et atomes (on dit aussi variables propositionnelles, lettres) :* $\Vdash_0$ *est un sous ensemble de* $\mathcal{M} \times \mathsf{P}$*. Comme souvent pour une relation binaire, on notera* $\mathfrak{m}_i \Vdash p_k$ *pour* $(\mathfrak{m}_i, p_k)$ *appartient à* $\Vdash_0$.

Remarque 1. *On obtient des définitions équivalentes de modèle de Kripke (par rapport à la signification des formules) en remplaçant dans la définition ci-dessous :*

1. *la relation $\Vdash_0$, par une fonction L (dite fonction d'étiquetage) qui associe à chaque monde m un ensemble (éventuellement vide) de variable(s) propositionnelle(s) ou bien,*
2. *la relation $\Vdash_0$ par une fonction V (dite fonction d'évaluation) qui associé à chaque lettre propositionnelle une ensemble (éventuellement vide) de mondes.*

Dans la suite, nous utiliserons parfois une de ce deux définitions alternative lorsqu'elle est plus pratique.

2.3 Interprétation des formules

Définition 3 (Vérité dans un modèle)**.** *Soit $< \mathcal{M}, \mathrm{R}, \Vdash_0 >$ un modèle de Kripke. Nous étendons la relation $\Vdash_0$ à toutes les formules du langage, en définissant $\mathfrak{m}_i \Vdash A$ par induction sur la formule A.*

1. *lorsque $p \in \mathcal{P}$ est une variable propositionnelle $\mathfrak{m}_i \Vdash p$ si et seulement si $\mathfrak{m}_i \Vdash_0 p$;*
2. *$\mathfrak{m}_i \Vdash \neg A$ si et seulement si $\mathfrak{m}_i \nVdash A$;*
3. *$\mathfrak{m}_i \Vdash A \wedge B$ si et seulement si ($\mathfrak{m}_i \Vdash A$ et $\mathfrak{m}_i \Vdash B$) ;*
4. *$\mathfrak{m}_i \Vdash A \rightarrow B$ si et seulement si ($\mathfrak{m}_i \Vdash A$ implique $\mathfrak{m}_i \Vdash B$) ;*
5. *$\mathfrak{m}_i \Vdash A \vee B$ si et seulement si ($\mathfrak{m}_i \Vdash A$ ou $\mathfrak{m}_i \Vdash B$) ;*
6. *$\mathfrak{m}_i \Vdash \Box A$ si et seulement si $\mathfrak{m}_j \Vdash A$ pour tout j tel que $\mathfrak{m}_i \mathrm{R} \mathfrak{m}_j$;*
7. *$\mathfrak{m}_i \Vdash \Diamond A$ si et seulement si $\mathfrak{m}_j \Vdash A$ pour au moins un j tel que $\mathfrak{m}_i \mathrm{R} \mathfrak{m}_j$.*

Remarque 2. *Si nous utilisons les définitions alternatives de modèle de Kripke présentes dans la remarque 1, la clause de vérité dans un modèle pour une formule atomique devient, respectivement :*

— $\mathfrak{M}, \mathfrak{m}_i \Vdash p$ *si et seulement si* $p \in L(\mathfrak{m}_i)$
— $\mathfrak{M}, \mathfrak{m}_i \Vdash p$ *si et seulement si* $\mathfrak{m}_i \in V(p)$

Si $\mathfrak{m}_i \Vdash A$ nous dirons que A est vrai en $\mathfrak{m}_i$, ou que $\mathfrak{m}_i$ force ou valide A. S'il est faux que $\mathfrak{m}_i \Vdash A$ alors nous écrirons $\mathfrak{m}_i \nVdash A$ et nous dirons que $\mathfrak{m}_i$ ne force pas A ou encore que A est fausse dans $\mathfrak{m}_i$.

Dans un modèle de Kripke la définition de vérité pour les connecteurs binaires est tout à fait identique à la définition de vérité utilisant les tables de vérité du calcul propositionnel usuel. En effet, le lecteur peut aisément vérifier les équivalences standard entre connecteurs, par exemple $\mathfrak{m}_i \Vdash A \rightarrow B$ équivaut à $\mathfrak{m}_i \Vdash \neg(A \wedge \neg B)$.

On remarquera que pour savoir si $\mathfrak{m}_i \Vdash A$, c'est-à-dire si une formule A est vraie dans un monde i, il faut savoir si d'autres formules sont vraies dans d'autres mondes $\mathfrak{m}_j \Vdash A'$ (A' est une sous formules de A et les $\mathfrak{m}_j$ sont les mondes accessibles à partir de $\mathfrak{m}_j$). C'est clairement plus compliqué que de savoir si une formules complexes est vraie ou fausse pour une valuation donnée sur les atomes dans le cas habituel du calcul propositionnel classique.

Par la suite nous ne ferons plus la différence entre $\Vdash_0$ et $\Vdash$, vu que $\Vdash_0$ définit $\Vdash$ sans équivoque et coïncide sur les variables propositionnelles, et que $\Vdash$ contient $\Vdash_0$.

2.4 Validité d'une formule dans un modèle

Soit A une formule, nous dirons que

— A **est valide dans le modèle** $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$, ce qui sera noté $\mathfrak{M} \models A$ lorsque $\mathfrak{m}_i \Vdash A$ pour tout monde $\mathfrak{m}_i$ de $\langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$.

— A **est valide dans le cadre** $C = \langle \mathcal{M}, R \rangle$ si $\mathfrak{M} \models A$ pour tout modèle $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$ basé sur C, c'est-à-dire pour toute relation $\Vdash_0$

— et si L est une collection de cadres, **la formule** A **est valide dans les cadres de** L si elle est valide dans chaque cadre de L.

2.5 Schémas d'axiomes et propriétés du cadre

Notation 3. *Étant données*

— *une formule ϕ comportant k variables propositionnelles $p_1, p_2, \ldots, p_k$*

— *k formules $G_1, \ldots, G_k$*

on note

$$\phi[G_1/p_1, G_2/p_2, \ldots, G_k/p_k]$$

le résultat de la substitution simultanée *de G_i à chaque occurrence de p_i dans ϕ pour $1 \leqslant i \leqslant k$ — on peut aussi dire du remplacement simultané de chaque occurrence de p_i par G_i pour $1 \leqslant i \leqslant k$.*

Exemple 4. *Si*

— $\phi = \Box(p_1 \to p_2) \to \Box p_1 \to \Box p_2$,

— $G_1 = p_1 \to p_2$

— $G_2 = p_1$

alors

$$\phi[G_1/p_1, G_2/p_2] = \Box(\underline{(p_1 \to p_2)} \to \underline{\underline{p_1}}) \to \Box\underline{(p_1 \to p_2)} \to \Box\underline{\underline{p_1}}$$

*On remarque l'importance de l'adjectif "*simultanée*" : il ne faut pas remplacer p_1 par G_1 puis p_2 par G_2 (ce qui donnerait $\Box((p_1 \to p_1) \to p_1) \to \Box(p_1 \to p_1) \to \Box p_1$) ni remplacer p_2 par G_2 puis p_1 par G_1 (ce qui donnerait $\Box((p_1 \to p_2) \to (p_1 \to p_2) \to \Box(p_1 \to p_2) \to \Box(p_1 \to p_2)$). Il faut noter les occurrences de p_1 et de p_2 et écrire G_1 à la place de chaque occurrence de p_1 et G_2 à la place de chaque occurrence de p_2, ce qui donne comme indiqué $\Box((p_1 \to p_2) \to p_1) \to \Box(p_1 \to p_2) \to \Box p_1$.*

Un schéma d'axiomes ϕ est une formule propositionnelle, mais qui est vue comme l'ensemble de des instances du schéma d'axiomes. [1]

Définition 4. *Un schéma d'axiomes ϕ est tout simplement une formule, mais qui est vue comme l'ensemble de ses instances. Si ϕ utilise les variables propositionnelles $p_1, p_2, \ldots, p_k$ une instance du schéma d'axiome ϕ est une formule $\phi[G_1/p_1, G_2/p_2, \ldots, G_k/p_k]$ où $G_1, \ldots, G_k$ sont des formules propositionnelles (cf. supra).*

1. On parle d'*instance*, car le schéma d'axiomes ϕ peut être vu comme une formule du second ordre $\forall^2 p_1 \forall^2 p_2 \cdots \forall^2 p_k . \phi$ où $\forall^2$ est la quantification sur les variables propositionnelle (à distinguer de la quantification du premier ordre). Bien sûr un renommage des variables propositionnelles liées par ces $\forall^2$ est possible et les différentes variables propositionnelles peuvent être instanciées indépendamment.

Définition 5. *Si ϕ est un schéma d'axiomes qui dépend des variables propositionnelles $p_1, p_2, \ldots, p_k$ nous dirons que ϕ est valide dans le modèle de Kripke $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, \Vdash \rangle$ (resp. dans le cadre $\langle \mathcal{M}, \mathrm{R} \rangle$) si et seulement* toute *instance $\phi[G_1/p_1, G_2/p_2, \ldots, G_k/p_k]$ de ϕ (comme défini supra) est valide dans $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, \Vdash \rangle$. Un schéma d'axiomes est valide dans un cadre $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}$ lorsque chacune de ses instances est valide dans tout modèle basé sur ce cadre. Un schéma d'axiome est valide dans une famille de cadre lorsque toute instance est valide dans tout modèle basée sur tout cadre de ladite famille de cadre.*

Comme la négation de *tout x a la propriété P* est *il existe au moins un x qui n'a pas la propriété P*, nous pouvons faire la remarque suivante :

Remarque 5. *Un schéma d'axiomes ϕ n'est pas valide dans un cadre $\langle \mathcal{M}, \mathrm{R} \rangle$, se et seulement s'il est possible de trouver*

- au moins une *instance $\phi[G_1/p_1, G_2/p_2, \ldots]$ de ϕ*
- au moins une *relation de forcing atomique $\Vdash_0$,*
- au moins un *monde $\mathfrak{m}_{i_0}$ de $\mathcal{M}$ tels que $\mathfrak{m}_i \not\Vdash \phi[G_1/p_1, G_2/p_2, \ldots]$*

Ainsi pour montrer qu'un schéma d'axiome ϕ est vrai dans un cadre $\langle \mathcal{M}, \mathrm{R} \rangle$ si et seulement si R a la propriété $\mathfrak{p}(R)$, il suffit de montrer que :

- si $R \subset \mathcal{M}^2$ a la propriété $\mathfrak{p}(R)$ alors
  pour tout forcing atomique $\Vdash$ défini sur $\mathcal{M}$
  pour toute instance $\phi[G_1/p_1, G_2/p_2, \ldots]$ de ϕ
  et pour tout monde $\mathfrak{m}_i$ de $\mathcal{M}$
  on a $\mathfrak{m}_i \Vdash \phi[G_1/p_1, G_2/p_2, \ldots]$;
- si $R \subset \mathcal{M}^2$ n'a pas la propriété $\mathfrak{p}(R)$ alors il existe
  une instance $\phi[G_1/p_1, G_2/p_2, \ldots]$ de ϕ,
  une relation de forcing atomique $\Vdash$,
  et un monde $\mathfrak{m}_i$ de $\mathcal{M}$
  tels que $\mathfrak{m}_i \not\Vdash \phi[G_1/p_1, G_2/p_2, \ldots]$

Théorème 6. *Le schéma d'axiome*

$$(\kappa) : \quad \Box(p_1 \to p_2) \to \Box p_1 \to \Box p_2$$

est vrai dans tout modèle de Kripke.

Démonstration. cf exercices en section 2.8 □

Définition 6. *Un cadre $\langle \mathcal{M}, \mathrm{R} \rangle$ est dit réflexif, transitif, symétrique, euclidien, sériel etc. lorsque la relation* R *a la propriété correspondante.*

Rappelons ces définitions :

Définition 7.

- *une relation* R *est dite* réflexive *lorsque pour tout x du domaine $x \mathrm{R} x$*
- *une relation* R *est dite* transitive *lorsque pour tout x, y, z du domaine si $x \mathrm{R} y$ et $y \mathrm{R} z$ alors $x \mathrm{R} z$.*

— *une relation* R *est dite* symétrique *lorsque pour tous* x, y *du domaine si* $x \mathrm{R} y$ *alors* $x \mathrm{R} y$
— *une relation* R *est dite* euclidienne *lorsque pour tous* x, y, z *du domaine tels que* $x \mathrm{R} y$ *et* $x \mathrm{R} z$ *on a* $y \mathrm{R} z$ *(et donc aussi* $z \mathrm{R} y$*, puisque les rôles de* y *et de* z *sont symétriques). Bien évidemment il est possible que certains de* x, y, z *soient identiques, par exemple* xRx *et* xRy *entraîne* yRx*, et aussi* xRy *entraîne* yRy*...*
— *une relation* R *est dite* dense *lorsque pour tous* x, y *du domaine tels que* $x \mathrm{R} y$ *il existe un élément* z *du domaine tel que* $x \mathrm{R} z$ *et* $z \mathrm{R} y$*.*
— *une relation* R *est dite* sérielle *lorsque pour tout* x *du domaine il existe un élément* z *du domaine tel que* $x \mathrm{R} z$

Théorème 7. *Le schéma d'axiome* nom *décrit par la formule* axiome *est satisfait dans toute interprétation de cadre* $\langle (\mathfrak{m}_i)_{i \in I}, \mathrm{R} \rangle$ *si et seulement si l'accessibilité* R *est une relation ayant la propriété indiqué dans le tableau que voici :*

nom	*axiome*	accessibilité
T	$\Box p \to p$	*réflexive*
4	$\Box p \to \Box \Box p$	*transitive*
B	$p \to \Box \Diamond p$	*symétrique*
5	$\Diamond p \to \Box \Diamond p$	*euclidienne*
D	$\Box p \to \Diamond p$	*sérielle*
	$\Box \Box p \to \Box p$	*dense*

Démonstration. cf exercices en section 2.8 □

2.6 Conséquence logique et modèles de Kripke

Il faut distinguer deux sortes de conséquence sémantique : locale et globale.

2.7 Bissimulation

Définition 8 (Bissimulation)**.** *Soient* $\mathcal{M}^a = \langle \mathfrak{m}_i^a, R^a, \Vdash_0^a \rangle$ *et* $\mathcal{M}^b = \langle \mathfrak{m}_i^b, R^b, \Vdash_0^b \rangle$ *deux modèles de Kripke. Une relation binaire B entre* $(\mathfrak{m}_i^a)_{i \in I}$ *et* $(\mathfrak{m}_j^b)_{j \in J}$ *est une bissimulation lorsque :*
— *Pour toute variable propositionnelle p, pour tous mondes* $\mathfrak{m}_i^a$ *et* $\mathfrak{m}_j^b$ *tels que* $\mathfrak{m}_i^a B \mathfrak{m}_j^b$*,* $\mathfrak{m}_i^a \Vdash_0^a p$ *si et seulement si* $\mathfrak{m}_j^b \Vdash_0^b p$*.*
— *Pour tous mondes* $\mathfrak{m}_i^a$ *et* $\mathfrak{m}_j^b$ *tels que* $\mathfrak{m}_i^a B \mathfrak{m}_j^b$*, s'il existe un monde* $\mathfrak{m}_{i'}^a$ *de* $\mathcal{M}^a$ *tel que* $\mathfrak{m}_i^a R^a \mathfrak{m}_{i'}^a$ *alors il existe un monde* $\mathfrak{m}_{j'}^b$ *de* $\mathcal{M}^b$ *tel que* $\mathfrak{m}_{i'}^a B \mathfrak{m}_{j'}^b$ *et* $\mathfrak{m}_j^b R^b \mathfrak{m}_{j'}^b$*.*
— *Pour tous mondes* $\mathfrak{m}_i^a$ *et* $\mathfrak{m}_j^b$ *tels que* $\mathfrak{m}_i^a B \mathfrak{m}_j^b$*, s'il existe un monde* $\mathfrak{m}_{j'}^b$ *de* $\mathcal{M}^b$ *tel que* $\mathfrak{m}_j^b R^b \mathfrak{m}_{j'}^b$*, alors il existe un monde* $\mathfrak{m}_{i'}^a$ *de* $\mathcal{M}^a$ *tel que* $\mathfrak{m}_{i'}^a B \mathfrak{m}_{j'}^b$ *et* $\mathfrak{m}_i^a R^a \mathfrak{m}_{i'}^a$*.*

Théorème 8. *Etant donnés*
— *Une relation de bisimulation B entre deux modèles de Kripke* $\mathcal{M}^a = \langle \mathfrak{m}_i^a, R^a, \Vdash_0^b \rangle$ *et* $\mathcal{M}^b = \langle \mathfrak{m}_i^a, R^a, \Vdash_0^b \rangle$

— *deux mondes* $\mathfrak{m}_i^a$ *dans* $\mathcal{M}^a$ *et* $\mathfrak{m}_j^b$ *dans* $\mathcal{M}^b$ *bissimilaires c'est-à-dire satisfaisant* $\mathfrak{m}_i^a B \mathfrak{m}_j^b$

pour toute formule G on a $\mathfrak{m}_i^a \Vdash G$ *si et seulement si* $\mathfrak{m}_j^b \Vdash G$.

Démonstration. **Cette preuve est un excellent exercice pour s'apprendre à raisonner par induction sur les formules.** On procède par induction sur la formule G.

— Si G est une variable propositionnelle, c'est vrai par définition de la bissimilarité : pour tous mondes $\mathfrak{m}_i^a$ et $\mathfrak{m}_j^b$ tels que $\mathfrak{m}_i^a B \mathfrak{m}_j^b$, $\mathfrak{m}_i^a \Vdash_0^a p$ si et seulement si $\mathfrak{m}_j^b \Vdash_0^b p$.
— Si G est de la forme $\neg H$...
— Si G est de la forme $H \to H'$...
— Si G est de la forme $\Diamond H$...

Cela suffit, car les autres connecteurs $\&, \vee$ et $\Box$ sont définissables à partir de $\neg, \to, \Diamond$ □

Remarque 9. *Au lieu de traiter le cas ou* $G = \Box H$ *plutôt que* $G = \Diamond H$*, c'est plus amusant.*

2.8 Exercices classiques sur les modèles de Kripke

Exercice 10. *Montrer que le schéma d'axiome*

$$(\kappa) : \Box(A \to B) \to \Box A \to \Box B$$

est valide dans tout modèle de Kripke.

Solution: Soit $\langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$ un modèle et soit $\mathfrak{m}_i \in \mathcal{M}$. Supposons que $\mathfrak{m}_i \Vdash \Box(A \to B)$. Par définition ceci veut dire que $\mathfrak{m}_j \Vdash A \to B$ pour tout j tel que $\mathfrak{m}_i \mathrm{R} \mathfrak{m}_j$. Supposons que $\mathfrak{m}_i \Vdash \Box A$. Il faut maintenant montrer que $\mathfrak{m}_i \Vdash \Box B$. Ceci est immédiat : pour tout j tel que $\mathfrak{m}_i \mathrm{R} \mathfrak{m}_j$ on obtient que $\mathfrak{m}_j \Vdash A$ et $\mathfrak{m}_j \vdash A \to B$, et donc que $\mathfrak{m}_j \vdash B$. On en conclut que $\mathfrak{m}_i \Vdash \Box B$.

Exercice 11. *Montrer que le schéma d'axiome $\Box A \to A$ est valide dans un cadre $\langle \mathcal{M}, \mathrm{R} \rangle$ si et seulement si la relation d'accessibilité* R *est réflexive. (cf section 2.5 du cours)*

Solution:

1. Toutes les instances de l'axiome $\Box A \to A$ sont valides dans tout monde de toute interprétation $\langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$ lorsque R réflexive.

   Montrons que $\mathfrak{m}_j \Vdash \Box A \to A$, c'est-à-dire que si $\mathfrak{m}_j \Vdash \Box A$ alors $\mathfrak{m}_j \Vdash A$. Dire que $\mathfrak{m}_j \Vdash \Box A$ c'est dire que pour tout j tel que $\mathfrak{m}_j \mathrm{R} \mathfrak{m}_k$ on a $\mathfrak{m}_k \Vdash A$, et comme R est réflexive on a $\mathfrak{m}_j R \mathfrak{m}_j$ et donc $\mathfrak{m}_j \Vdash A$. Par conséquent si $\mathfrak{m}_j \Vdash \Box A$ alors $\mathfrak{m}_j \Vdash A$, c'est-à-dire $\mathfrak{m}_j \Vdash \Box A \to A$.

2. Si R n'est pas réflexive il est possible de trouver une interprétation de $\Vdash_0$ de cadre $\langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$ telle que une instance du schéma d'axiomes $\Box A \to A$ soit fausse. Comme R n'est pas réflexive, il existe un monde $\mathfrak{m}_{j_0}$ tel que $\mathfrak{m}_{j_0} \not\mathrm{R} \mathfrak{m}_{j_0}$. Pour ce $\mathfrak{m}_{j_0}$ et pour une proposition atomique p définissons $\Vdash_0$ ainsi : $\mathfrak{m}_{j_0} \not\Vdash p$ et quel que soit k tel que $\mathfrak{m}_{j_0} \mathrm{R} \mathfrak{m}_k$ on a $\mathfrak{m}_k \Vdash p$ (c'est possible car $\mathfrak{m}_{j_0} \not\mathrm{R} \mathfrak{m}_{j_0}$). On a alors $\mathfrak{m}_{j_0} \Vdash \Box A$ et $\mathfrak{m}_{j_0} \not\Vdash A$ c'est-à-dire $\mathfrak{m}_{j_0} \not\Vdash \Box A \to A$

Exercice 12. *Montrer que le schéma d'axiome $\Box A \to \Box\Box A$ est valide dans un cadre $\langle \mathcal{M}, \mathrm{R} \rangle$ si et seulement si la relation d'accessibilité* R *est transitive. (cf section 2.5 du cours.)*

Solution:

1. Soit une interprétation avec une une relation d'accessibilité transitive. Montrons que pour tout monde $\mathfrak{m}_j$ on a $\mathfrak{m}_j \Vdash \Box A \to \Box\Box A$. Il faut pour cela montrer que si $\mathfrak{m}_j \Vdash \Box A$ alors $\mathfrak{m}_j \Vdash \Box\Box A$.

   Ce qu'il faut établir, c'est que $\mathfrak{m}_j \Vdash \Box\Box A$ c'est-à-dire que pour tout monde $\mathfrak{m}_k$ tel que $\mathfrak{m}_j \mathrm{R} \mathfrak{m}_k$ on a $\mathfrak{m}_k \Vdash \Box A$, c'est-à-dire pour tout monde $\mathfrak{m}_l$ tel que $\mathfrak{m}_k \mathrm{R} m_l$ on a $\mathfrak{m}_l \Vdash A$.

   Si $\mathfrak{m}_j R \mathfrak{m}_k$ et $\mathfrak{m}_k R \mathfrak{m}_l$ comme R transitive on a $\mathfrak{m}_j R \mathfrak{m}_l$, et comme $\mathfrak{m}_j R \mathfrak{m}_l$ on a $\mathfrak{m}_l \Vdash A$, ce qu'il fallait démontrer.

2. Soit maintenant un cadre non transitif. Alors il existe trois mondes tels que $\mathfrak{m}_{j_0}\mathrm{R}\mathfrak{m}_{k_0}$ et $\mathfrak{m}_{k_0}\mathrm{R}\mathfrak{m}_{l_0}$ sans que $\mathfrak{m}_{j_0}\mathrm{R}\mathfrak{m}_{l_0}$. En posant $\mathfrak{m}_k \Vdash A$ dès que $\mathfrak{m}_{j_0}\mathrm{R}\mathfrak{m}_k$ et $\mathfrak{m}_{l_0} \nVdash A$, on a $\mathfrak{m}_{j_0} \Vdash \Box A$ and $\mathfrak{m}_{j_0} \nVdash \Box\Box A$ et donc $\mathfrak{m}_{j_0} \nVdash \Box A \rightarrow \Box\Box A$

Exercice 13. *Montrer que le schéma d'axiomes $A \rightarrow \Box\Diamond A$ est valide dans un cadre $\langle \mathcal{M}, \mathrm{R}\rangle$ si et seulement si la relation d'accessibilité* R *est symétrique. (cf section 2.5 du cours.)*

Exercice 14. *Montrer que le schéma d'axiomes $\Box\Box A \rightarrow \Box A$ est valide dans un cadre $\langle \mathcal{M}, \mathrm{R}\rangle$ si et seulement si la relation d'accessibilité* R *est dense. (cf section 2.5 du cours.)*

Exercice 15. *Montrer que le schéma d'axiomes $\Box A \rightarrow \Diamond A$ est valide dans un cadre $\langle \mathcal{M}, \mathrm{R}\rangle$ si et seulement si la relation d'accessibilité* R *est sérielle. (cf section 2.5 du cours.)*

Exercice 16. *Montrer que le schéma d'axiomes $\Diamond A \rightarrow \Box\Diamond A$ est valide dans un cadre $\langle \mathcal{M}, \mathrm{R}\rangle$ si et seulement si la relation d'accessibilité* R *est euclidienne. (cf section 2.5 du cours.)*

Solution:

1. Soit $C = \langle \mathcal{M}, R\rangle$ un cadre euclidien quelconque et soit $\langle \mathcal{M}, \mathrm{R}, \Vdash_0\rangle$ un modèle basé sur C. Soit $\mathfrak{m}_i \in \mathcal{M}$ et supposons que $\mathfrak{m}_i \Vdash \Diamond A$. Comme $\mathfrak{m}_i \Vdash \Diamond A$ il existe au moins un monde disons $\mathfrak{m}_{k_0}$, qui est accessible depuis $\mathfrak{m}_i$ c'est-à-dire $\mathfrak{m}_i R \mathfrak{m}_{k_0}$ et qui satisfait A c'est-à-dire $\mathfrak{m}_{k_0} \Vdash A$. Pour montrer que $\mathfrak{m}_i \Vdash \Box\Diamond A$, il faut montrer que pour *tout* monde $\mathfrak{m}_j$ accessible à partir de $\mathfrak{m}_i \Vdash \Diamond A$, c'est-à-dire qu'il existe un monde $\mathfrak{m}_l$ accessible à partir de $\mathfrak{m}_j$ tel que $\mathfrak{m}_l \Vdash A$. Comme la relation est euclidienne et que $\mathfrak{m}_i R \mathfrak{m}_j$ et $\mathfrak{m}_i R \mathfrak{m}_{k_0}$ on a $\mathfrak{m}_j R \mathfrak{m}_{k_0}$. Donc de tout monde $\mathfrak{m}_j$ accessible à partir de $\mathfrak{m}_i$ il y a un monde accessible à partir de $\mathfrak{m}_j$ où A est vrai, à savoir le monde $\mathfrak{m}_{k_0}$, c'est-à-dire $\mathfrak{m}_i \Vdash \Box\Diamond A$, qed.
2. Soit un cadre constitué de deux mondes $\mathfrak{m}_0$ et $\mathfrak{m}_1$ avec l'accessibilité $\mathfrak{m}_0 R \mathfrak{m}_1$ et $\mathfrak{m}_0 R \mathfrak{m}_0$ — qui n'est pas euclidienne, car on a $\mathfrak{m}_0 R \mathfrak{m}_1$ et $\mathfrak{m}_0 R \mathfrak{m}_0$ sans avoir $\mathfrak{m}_1 R \mathfrak{m}_0$. Supposons que p soit vraie en $\mathfrak{m}_1$ mais pas en $\mathfrak{m}_0$. On a $\mathfrak{m}_0 \Vdash \Diamond p$ (car $\mathfrak{m}_1 \Vdash p$). Comme il n'y a pas de monde accessible depuis $\mathfrak{m}_1$, on a $\mathfrak{m}_1 \nVdash \Diamond p$, et donc comme $\mathfrak{m}_0 R \mathfrak{m}_1$, on a $\mathfrak{m}_0 \nVdash \Box\Diamond p$. Et donc $\mathfrak{m}_0 \nVdash \Diamond p \multimap \Box\Diamond p$.

Exercice 17. *Soit $\langle \mathcal{M}, \mathrm{R}, \Vdash_0\rangle$ un modèle réflexif et soit $\mathfrak{m}_i$ un monde dans $\mathcal{M}$. Peut-on avoir*

$$\begin{array}{ll} \mathfrak{m}_i & \Vdash P \\ \mathfrak{m}_i & \Vdash \Box Q \rightarrow \Box\neg P \\ \mathfrak{m}_i & \Vdash \Box Q \end{array}$$

où P et Q sont deux lettres propositionnelles. Justifiez votre réponse.

Exercice 18. *Soit* KT5 *la collections des cadres qui sont réflexifs et euclidiens et soit* KT4B *la collections des cadres qui sont réflexifs, transitifs et symétriques. Montrer que les formules valides dans tous les modèles de cadre KT5 sont les mêmes qui sont valides dans les cadres de KTB4.*

Démonstration. Il suffit de montrer que toute relation réflexive et euclidienne est transitive et symétrique et que, vice-versa toute relation transitive, réflexive et symétrique est euclidienne. □

2.9 Exercice : somme disjointe de modèles de Kripke

Soit $\mathcal{M}^p = \langle (\mathfrak{m}_i^p)_{i\in I^p}, R^p, \Vdash_0^p \}$, pour $p \in J$ une famille de modèles de Kripke.
On définit un nouveau modèles de Kripke

$$\mathcal{M} = \uplus_{p\in J}\mathcal{M}^p = \langle (\mathfrak{m}_i^p)_{p\in J, i\in I^p}, R = \cup_{p\in J} R^p, \Vdash^0 = \cup_{p\in J} \Vdash_0^p \rangle$$

soit, en français, le modèle de Kripke $\mathcal{M}$ est la juxtaposition (l'union disjointe) des modèles de Kripke $\mathcal{M}^p$:

les mondes possibles sont la réunion des mondes possibles,

deux mondes sont en relation si et seulement s'ils proviennent tous les deux du même modèle M^p et s'ils étaient en relation dans M^p,

et un monde $\mathfrak{m}$ de $\mathcal{M}$ force la variable propositionnelle a, $\mathfrak{m} \Vdash_0^p a$, si et seulement si ce monde $\mathfrak{m}$ provient du modèle $\mathcal{M}^p$ et si dans $\mathcal{M}^p$ on avait $\mathfrak{m} \Vdash_0^p a$.

1. Soient les deux modèles de Kripke
   $\mathcal{M}^1 = \langle \{\mathfrak{m}_1^1, \mathfrak{m}_2^1\}; R^1 = (\mathfrak{m}_1^1, \mathfrak{m}_2^1); \Vdash_0^1 = \{\mathfrak{m}_1^1 \Vdash_0^1 a, \mathfrak{m}_2^1 \Vdash_0 a, \mathfrak{m}_2^1 \Vdash_0 b\}\rangle$.
   $\mathcal{M}^2 = \langle \{\mathfrak{m}_1^2\}; R^2 = (\mathfrak{m}_1^2, \mathfrak{m}_1^2); \Vdash_0^2 = \{\mathfrak{m}_1^2 \Vdash_0^2 b\}\rangle$.
   (a) Dessinez $\mathcal{M}^1$ et montrer que $\mathfrak{m}_1^1 \Vdash^1 \Box a$
   (b) Dessinez $\mathcal{M}^2$ et montrez que $\mathfrak{m}_1^2 \Vdash^2 \Box b$.
   (c) Dessinez $\mathcal{M}^{12} = \mathcal{M}^1 \uplus \mathcal{M}^2$ et montrez que on a $\mathfrak{m}_1^1 \Vdash^{12} \Box a$ ($\Vdash^{12}$ désigne le forcing dans $\mathcal{M}^{12} = \mathcal{M}^1 \uplus \mathcal{M}^2$).
2. Soit $\mathcal{M}^p = \langle (\mathfrak{m}_i^p)_{i\in I^p}, R^p, \Vdash_0^p \rangle$, pour $p \in J$ une famille de modèles de Kripke et $\mathcal{M} = \uplus_{p\in J}\mathcal{M}^p$ comme défini ci-dessus. Soit $\mathfrak{m}_i^p$ un monde du modèle $\mathcal{M}^p$ et donc de $\mathcal{M}$. Montrez que pour toute formule H, $\mathfrak{m}_i^p \Vdash^p H$ si et seulement si $\mathfrak{m}_i^p \Vdash H$ ($\Vdash$ désigne le forcing dans $\mathcal{M} = \uplus_{p\in J}\mathcal{M}^p$). On procédera par induction sur la formule H. On étudiera le cas de base (H est une variable propositionnelle), le cas où H est une implication, le cas où H est une négation, et le cas où H est une formule nécessaire ($\Box$). [Bien préciser quand vous utilisez l'hypothèse d'induction. Pour le cas de $\Box$ il faudra faire attention à distinguer l'accessibilité R dans $\mathcal{M}$ et l'accessibilité R^p dans l'un des $\mathcal{M}^p$.
3. On souhaite définir une modalité A dans les modèles de Kripke par $\mathfrak{m}_i \Vdash \mathsf{A}G$ si et seulement si $\mathfrak{m} \Vdash G$ pour tout monde $\mathfrak{m}$ de ce modèle.
   (a) Montrer en reprenant les exemples $\mathcal{M}^1$, $\mathcal{M}^2$ et $\mathcal{M}^{12}$ ci-dessus que $\mathfrak{m}_1^1 \Vdash^1 \mathsf{A}a$ et que $\mathfrak{m}_1^1 \nVdash^{12} \mathsf{A}a$ ($\Vdash^{12}$ désigne le forcing dans $\mathcal{M}^{12} = \mathcal{M}^1 \uplus \mathcal{M}^2$).

(b) En déduire que $\mathsf{A}G$ ne peut pas s'exprimer par une formule de la logique modale habituelle $\Box, \rightarrow, \neg$.

4. Expliquer en quelques lignes pourquoi c'est un cas de bissimulation. On donnera les deux modèles en bissimulation et on précisera la relation de bissimulation.

Chapitre 3

Complétude des systèmes déductifs à la Hilbert par rapport aux modèles de Kripke

3.1 Systèmes déductifs à la Hilbert

La définition des schémas d'axiomes a été donnée dans la définition 4 du chapitre 2.

Définition 9. *Un système déductif à la Hilbert est défini par des schémas d'axiomes et par des règles d'inférence. Une preuve est une suite de formules, chacune étant*

- *soit une instance d'un schéma d'axiomes,*
- *soit déduite par une règle d'inférence à partir d'une ou plusieurs lignes précédentes.*

Les tautologies de la logique ainsi décrite sont toutes les formules qui figurent dans une preuve.

Pour la logique intuitionniste, les schémas d'axiomes sont :

S : $(p \to (q \to r)) \to (p \to q) \to (p \to r)$

K : $p \to (q \to p)$

$\bot$: $\bot \to C$ for all C.

$\neg A$ est une abréviation pour $A \to \bot$

$\wedge$: $p \to (q \to (p \wedge q))$

π_1 : $(p \wedge q) \to p$

π_2 : $(p \wedge q) \to q$

$\vee_1$: $p \to (p \vee q)$

$\vee_2$: $q \to (p \vee q)$

$\vee_e$: $(p \vee q) \to (p \to r) \to (q \to r) \to r$

Et l'unique règle de déduction est le modus ponens : si une ligne est de la forme $A \to B$ et une autre ligne de la forme A alors on peut ajouter B à la preuve.

Pour obtenir la logique classique, il suffit de rajouter le schéma d'axiome : $((p \to \bot) \to \bot) \to p$. La disjonction $A \vee B$ est une abréviation pour $\neg(\neg A \wedge \neg B)$. Une variante assez courante de la double négation supra est la forme suivante du raisonnement par l'absurde : $(\neg a \to b) \to (\neg a \to \neg b) \to a$.

Pour obtenir la logique modale il suffit d'ajouter l'axiome κ :

$(\kappa) : \Box(p \to q) \to \Box p \to \Box q$

Remarque 19. *On peut bien sûr instancier tous les schémas d'axiomes, (ceux de la logique classique ainsi que l'axiome* (κ)*) avec des formules modales comportant les symboles* $\Box$ *et* $\Diamond$*.*

Pour la logique modale, on ajoute une règle de déduction dite de nécessité (*necessitation*) : si une ligne de la preuve est A alors on peut écrire $\Box A$ en fin de preuve.

Remarque 20. *On pourrait comprendre — à tort ! — cette règle de nécessité comme* $A \to \Box A$ *et comme dans beaucoup de logiques modales on a* $\Box A \to A$ *cela voudrait dire que* $\Box A \leftrightarrow A$ *et quelle serait l'intérêt d'une telle modalité ?*

En fait, il faut comprendre la règle de nécessité ainsi : si A est une tautologie, alors $\Box A$ *est aussi une tautologie. C'est cohérent avec les modèles de Kripke : si A est vrai dans un modèle, et donc vrai dans tout monde possible, alors* $\Box A$ *est aussi vrai dans tout monde possible de ce modèle. Ainsi, si A est une tautologie et donc vrai dans tout modèle (en tout monde possible de ce modèle) il en va de même pour* $\Box A$*.*

Traiter quelques exemples de démonstrations formelles à la Hilbert.

3.2 Ensembles cohérents de formules

Définition 10. *Un ensemble de formules* Γ *est dit cohérent (en anglais,* consistent*) lorsque les formules de* Γ *ne permettent pas de dériver le faux noté* $\bot$*. Si une telle dérivation existe, elle n'utilise qu'un nombre fini de formule de* Γ*, par définition des dérivations.*

Lemme 21. *Tout ensemble de formules cohérent* Γ *peut être étendu en un ensemble de formules qui soit cohérent maximal (par rapport à l'inclusion pour cette propriété de cohérence).*

Démonstration. Il suffit de disposer d'une énumération des formules : $F_1, F_2, \ldots$ On pose $\Gamma_0 = \Gamma$ et ensuite $\Gamma_{n+1} = \Gamma_n \cup \{F_n\}$ si $\Gamma_n \cup \{F_n\}$ est cohérent, et $\Gamma_{n+1} = \Gamma_n$ sinon. Soit $\Gamma_\omega = \bigcup \Gamma_i$. Cet ensemble de formules contient Γ, est cohérent et maximal avec ces deux propriétés.

Γ_ω contient $\Gamma = \Gamma_0$.

Γ_ω est cohérent car si un ensemble de formules entraîne $\bot$ alors un ensemble fini de formules entraîne $\bot$ et cet ensemble fini de formules (qui intervient dans la preuve — finie — de $\bot$) est inclus dans l'un des Γ_i qui sont, par construction, cohérents.

Γ_ω est maximal à être cohérent et à contenir Γ. Soit maintenant B une formule telle que $\Gamma_\omega \cup \{B\}$ soit cohérent. Comme les F_i énumèrent toutes les formules possibles, $B = F_k$ pour un certain k. Comme $\Gamma_\omega \cup \{F_k\}$ cohérent, $\Gamma_{k+1} = \Gamma_k \cup \{F_k\}$ est cohérent, et donc $B = F_k \in \Gamma_{k+1} \subset \Gamma_\omega$.

□

Proposition 22. *Etant donné un ensemble de formules maximal cohérent* Γ.

1. *Si* $\Gamma \vdash A$ *alors* $A \in \Gamma$.
2. *Pour toute formule* B *on a soit* $B \in \Gamma$ *soit* $\neg B \in \Gamma$.

Démonstration. 1. Si $\Gamma \vdash A$ alors $\Gamma \cup \{A\}$ est tout aussi cohérent que Γ ; comme Γ est maximalement cohérent, nous avons forcément que $A \in \Gamma$.

2. Si $B \notin \Gamma$ (resp. $\neg B \notin \Gamma$) comme Γest maximalement cohérent, $\Gamma, B \vdash \bot$ (resp. $\Gamma, \neg B \vdash \bot$) donc $\Gamma \vdash \neg B$ (resp. $\Gamma \vdash B$) et donc $\neg B \in \Gamma$ (resp. $B \in \Gamma$).

□

Lemme 23. *Si*
$\{\neg\Box B, \Box A_1, \Box A_2, \ldots, \Box A_n, \ldots\}$ *est cohérent,*
alors
$\{\neg B, A_1, A_2 \ldots, A_n, \ldots\}$ *est aussi cohérent.*
On notera que B *joue un rôle particulier.*

Démonstration. On montre la contraposée. Si $\{\neg B, A_1, A_2 \ldots, A_n, \ldots\}$ n'est pas cohérent, alors l'incohérence provient d'un nombre fini de formules, et donc pour un certains k, $\neg B, A_1, A_2 \ldots, A_k \vdash \bot$, c'est-à-dire $A_1, A_2, \ldots, A_k \vdash B$, ou encore $\vdash (A_1 \wedge A_2 \wedge \cdots \wedge A_k) \rightarrow B$. Avec la règle de nécessité, on obtient $\vdash \Box((A_1 \wedge A_2 \wedge \cdots \wedge A_k) \rightarrow B)$ et avec l'axiome (κ) on obtient $\vdash \Box(A_1 \wedge A_2 \wedge \cdots \wedge A_k) \rightarrow \Box B$. De là, on peut obtenir $\vdash \Box(A_1 \wedge A_2 \wedge \cdots \wedge A_k) \rightarrow \neg\Box B \rightarrow \bot$. Comme $\Box(U \wedge W) \equiv (\Box U) \wedge (\Box W)$ on a $\vdash \Box A_1 \wedge \Box A_2 \wedge \cdots \wedge \Box A_k \rightarrow \neg\Box B \rightarrow \bot$ et donc
$\Box A_1 \wedge \Box A_2 \wedge \cdots \wedge \Box A_k \rightarrow \neg\Box B \vdash \bot$, c'est-à-dire $\{\neg\Box B, \Box A_1, \Box A_2, \ldots, \Box A_k\}$ non cohérent et donc $\{\neg\Box B, \Box A_1, \Box A_2, \ldots, \Box A_k, \ldots, A_n, \ldots\}$ non cohérent. □

3.3 Modèle canonique et lemme de la vérité

Définition 11 (Modèle canonique)**.** *Le modèle canonique* M_ℓ *est défini par :*

1. *Mondes possibles : les ensembles maximaux cohérents de formules* Γ_i.
2. *Accessibilité :* $\Gamma_i R \Gamma_j$ *lorsque pour toute formule* $\Box A \in \Gamma_i$ *on a* $A \in \Gamma_j$.
3. *Forcing atomique :* $\Gamma_i \Vdash p$ *si et seulement si* $p \in \Gamma_i$

Lemme 24 (Lemme de la vérité)**.** *Pour tout monde possible* Γ *du modèle canonique* M_ℓ *(*Γ *est un ensemble cohérent maximal de formules), et pour toute formule* A *on a*

$$\Gamma \Vdash A \qquad \textit{si et seulement si} \qquad A \in \Gamma$$

Démonstration. On procède par induction sur la formule A, c'est-à-dire on suppose l'équivalence vraie pour toute formule A' de hauteur inférieure à la hauteur de A et on montre que dans ces conditions l'équivalence est vraie pour A.

— $A = p$, p atomique : c'est la définition de $\Gamma_i \Vdash p$ pour p atomique.

— $A = B \rightarrow C$

1. Supposons que $\Gamma \Vdash B \rightarrow C$ et montrons que $(B \rightarrow C) \in \Gamma$. On peut donc supposer que si $\Gamma \Vdash B$ alors $\Gamma \Vdash C$, et par hypothèse d'induction on sait que $\Gamma \Vdash B$ équivaut à $B \in \Gamma$ et que $\Gamma \Vdash C$ équivaut à $C \in \Gamma$. Si $(B \rightarrow C) \notin \Gamma$ (*) comme Γ est maximalement cohérent, on a $(B \wedge \neg C) \in \Gamma$, et donc $B \in \Gamma$ et $\neg C \in \Gamma$, ce qui par hypothèse d'induction signifie $\Gamma \Vdash B$ et $\Gamma \nVdash C$, ce qui contredit (*), donc $B \rightarrow C \in \Gamma$
2. Supposons que $(B \rightarrow C) \in \Gamma$ et montrons que $\Gamma \Vdash B \rightarrow C$. Pour cela supposons que $\Gamma \Vdash B$ (c'est-à-dire $B \in \Gamma$ par hypothèse d'induction). Comme $B \rightarrow C \in \Gamma$ et $B \in \Gamma$ et que Γ est maximalement cohérent, $C \in \Gamma$, c'est-à-dire , par hypothèse d'induction, $\Gamma \Vdash C$

— Si $A = \neg B$

1. Si $\Gamma \Vdash \neg A$ par définition cela signifie que $\Gamma \nVdash A$. Par hypothèse d'induction (sur A) cela signifie que $A \notin \Gamma$, et d'après la proposition 22 on a $\neg A \in \Gamma$.
2. si $\neg A \in \Gamma$ d'après la proposition 22 on a $A \notin \Gamma$ et par hypothèse d'induction (sur A) on a $\Gamma \nVdash A$, c'est-à-dire $\Gamma \Vdash \neg A$.

— Si $A = \Box B$

1. Supposons que $\Box B \in \Gamma$ et montrons que $\Gamma \Vdash \Box B$. Soit Δ un monde possible (un ensemble cohérent maximal de formules) tel que $\Gamma R \Delta$. Par definition de R, on a $B \in \Delta$, et par hyptohèse d'induction $\Delta \Vdash B$. Comme cela vaut pour tout Δ tel que $\Gamma R \Delta$, on a $\Gamma \Vdash \Box B$
2. Supposons que $\Gamma \Vdash \Box B$ et montrons que $\Box B \in \Gamma$. On va plutôt montrer la contraposée : si $\Box B \notin \Gamma$ alors $\Gamma \nVdash \Box B$. Si $\Box B \notin \Gamma$ alors $\neg \Box B \in \Gamma$ d'après le lemme 22. Soit C_i les formules telles que $\Box C_i \in \Gamma$. On a donc $\{\neg \Box B, \Box C_1, \ldots, \Box C_n, \ldots\}$ cohérent. D'après le lemme 23 $\{\neg B, C_1, \ldots, C_n, \ldots\}$ est cohérent, et se complète en un ensemble cohérent maximal Π d'après le lemme 21. On a $\Gamma R \Pi$ car Π contient toutes les formules C_i telles que $\Box C_i \in \Gamma$. Comme $\neg B \in \Pi$, $B \notin \Pi$ et on n'a pas $\Pi \Vdash B$. Comme $\Gamma R \Pi$ et $\Pi \nVdash B$, $\Gamma \nVdash \Box B$.

□

Théorème 25. *si une théorie Th est cohérente alors il existe un modèles de Kripke $\mathcal{M}$ telle que chaque formule de H de Th soit vraie en au moins un noeud de $\mathcal{M}$.*

3.4 Théorème(s) de complétude

Un modèle $\mathcal{M}$ d'une théorie Th est un modèle de Kripke dans lequel chaque formule de Th est vraie en chaque monde possible.

Lemme 26. *Une formule est vraie dans tout monde possible de tout modèle de Th si et seulement si elle est démontrable à partir de Th.*

Démonstration. Que les formules démontrables à partir de Th soient valides dans tout modèle de Th (vraies en tout monde de tout modèle de Th) se vérifie aisément par induction sur la formule.

Pour montrer qu'une formule G est vraie dans tout monde satisfaisant Th dans un modèle quelconque, est démontrable à partir de Th, nous allons plutôt établir la contraposée : si une formule n'est pas démontrable à partir de Th, alors il existe un modèle dans lequel elle est fausse en au moins un des mondes de ce modèle, monde qui satisfait Th.

Si une formule X n'est pas démontrable à partir de Th, sa négation est cohérente avec Th, et cette négation $\neg X$ fait donc partie d'un ensemble maximalement cohérent $\Gamma_0 \supset Th \cup \{X\}$ qui est l'un des mondes possibles du modèle canonique. On a $\neg X \in \Gamma_0$, c'est-à-dire $\Gamma \Vdash \neg X$, via le lemme 24. On a donc $\Gamma \nVdash X$, X est fausse dans un monde du modèle canonique où Th est vraie, ce qu'il fallait démontrer. □

3.5 Complétude des logique modales avec axiomes sur les cadres

Ce résultat s'étend aisément aux logiques modales ayant des axiomes spécifiques D T 4 5 B etc. sous la forme suivante :

Théorème 27 (Complétude en présence d'axiomes)**.** *Une formules H est démontrable à partir des axiomes de la logique classique, de κ et ainsi que des axiomes $Ax_1, Ax_2, \ldots, Ax_n$ caractérisés chacun par une propriété P_i du cadre si et seulement si elle est vraie dans chaque modèle de Kripke dont le cadre satisfait $P_1, P_2, \ldots, P_n$.*

Démonstration. L'argument est le même mais il faut considérer les ensemble cohérents maximaux avec les axiomes de L et vérifier que le cadre du modèle canonique satisfait $P_1, P_2, \ldots, P_n$ (ou que le modèle canonique satisfait $Ax_1, Ax_2, \ldots, Ax_n$).

Nous laissons en exercice le fait que le modèle canonique quand il est construit en utilisant les axiomes $Ax_1, Ax_2, \ldots, Ax_n$ pour calculer la complétion d'une théorie satisfait $P_1, P_2, \ldots, P_n$.

Soit X une formule non démontrable. Sa négation $\neg X$ est cohérente. Le modèle canonique est un modèle qui satisfait les propriétés satisfait $P_1, P_2, \ldots, P_n$ et il contient un monde $\mathfrak{m}_{\neg X}$ qui est une théorie contenant $\neg X$. Donc X n'est pas vrai en tout monde de tout modèle satisfaisant les propriétés $P_1, P_2, \ldots, P_n$. □

Chapitre 4

Décidabilité

4.1 Introduction

Une question important concernant une logique est de savoir si elle est décidable, c'est-à-dire, si il existe une procédure algorithmique qui permet de répondre à la question "cette formule est-elle valide ?". La logique propositionnelle est décidable : en effet, nous pouvons tester si une formule est une tautologie en construisant une table de vérité. et pour une formule donnée, sa table de vérité est finie. Malheureusement, nous ne pouvons pas tester si une formule modale est vraie dans tous les modèles, car il y en a une infinité.

(TBC)

4.2 Préliminaires

Les *filtrations* nous permettent d'établir qu'un système de logique modale est décidable en montrant qu'il possède la *propriété des modèles finis*, c'est-à-dire que toute formule modale qui est vraie (ou fausse) dans un modèle (fini ou infini) est aussi vraie (ou fausse) dans un modèle *fini*. Les filtrations sont définies par rapport à des ensembles de formules qui sont clos par sou-formules. On définit cette notion comme suit.

Définition 12. *Un ensemble de formule* Γ *est* clos pas sou formules *lorsqu'il contient tous les sous formule d'une formule dans* Γ. *Autrement dit : si A appartient à* Γ *et B est un sou formule de A alors B appartient à* Γ *aussi. De plus, l'ensemble* Γ *est clos par modalité s'il est clos par sou formule et* $A \in \Gamma$ *implique* $\Box A, \Diamond A \in \Gamma$.

Un exemple simple d'ensemble qui est clos par sous formule est donné par l'ensemble des sous formule d'une certaine formule A. Dans la suite, nous utiliserons la définition alternative de modèle de Kripke dont la relation de forcing atomique $\Vdash_0$ est remplacé par une fonction V qui associe à chaque lettre propositionnelle un ensemble de mondes.

Définition 13. *Soit $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, V \rangle$ un modèle, et Γ un ensemble de formules qui est clos par sou formule. Nous définissons une relation binaire $\equiv_\Gamma \subseteq \mathcal{M} \times \mathcal{M}$ entre éléments de $\mathcal{M}$ par :*

$$m \equiv_\Gamma m' \text{ si et seulement si } \forall A \in \Gamma : \mathfrak{M}, m \Vdash A \iff \mathfrak{M}, m' \Vdash A$$

autrement dit : m et m' sont équivalents par rapport à Γ lorsqu'ils vérifient exactement les mêmes formules de Γ

Les lecteurs peuvent aisément vérifier la proposition suivante.

Proposition 28. *Pour tout modèle $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, V \rangle$, pour tout ensemble Γ qui est clos par sou formule : la relation $\equiv_\Gamma$ définie ci-dessous est une relation d'équivalence, c'est-à-dire, elle est* réflexive*,* symétrique *et* transitive*.*

Étant donné un modèle $\mathfrak{M}$, et un de ses mondes m, nous dénotons par $[m]_{\equiv_\Gamma}$ (ou simplement par $[m]$) la classe d'équivalence de m par rapport à $\equiv_\Gamma$, autrement dit $[m] = \{m' \in \mathcal{M} \mid m \equiv_\Gamma m'\}$.

4.3 Filtrations

Dans cette section, nous définissons les conditions qu'un modèle *doit* satisfaire pour être considéré une filtration d'un autre modèle par rapport à un ensemble de formules qui est clos par sou formule. Toute filtration $\mathfrak{M}^\star$ d'une modèle $\mathfrak{M}$ par rapport à un ensemble Γ a le même ensemble de mondes $\mathcal{M}^\star$ et la même fonction d'évaluation $V^\star$. Par contre, de filtrations différentes peuvent avoir des relations d'accessibilité différentes. Néanmoins, toutes ces relations d'accessibilité doivent respecter une certain nombre de contraintes.

Définition 14. *[Filtration] Soit $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, V \rangle$ un modèle et Γ un ensemble de formules qui est clos par sou formule. Un modèle $\mathfrak{M}^\star = \langle \mathcal{M}^\star, \mathrm{R}^\star, V^\star \rangle$ est une filtration de $\mathfrak{M}$ par rapport à Γ lorsque :*

1. *$\mathcal{M}^\star$ est l'ensemble de classe d'équivalence de $\mathcal{M}$ par rapport à $\equiv_\Gamma$, plus précisément $\mathcal{M}^\star = \{[m] \mid m \in \mathcal{M}\}$.*
2. *Pour tout m et m' appartenant à $\mathcal{M}$:*
   (a) *si $m \mathrm{R} m'$ alors $[m] \mathrm{R}^\star [m']$;*
   (b) *si $[m] \mathrm{R}^\star [m']$ alors pour tout $\Box A \in \Gamma$, si $\mathfrak{M}, m \Vdash \Box A$ alors $\mathfrak{M}, m' \Vdash A$;*
   (c) *si $[m] \mathrm{R}^\star [m']$ alors pour tout $\Diamond A \in \Gamma$, si $\mathfrak{M}, m' \Vdash A$ alors $\mathfrak{M}, m \Vdash \Diamond A$.*
3. *$V^\star(p) = \{[m] \mid m \in V(p)\}$*

Nous pouvons maintenant démontrer le théorème exprimant la propriété fondamentale des filtrations qui va être utilisé dans la suite pour montrer la décidabilité des logiques modales.

Théorème 29. *Soit $\mathfrak{M}$ un modèle et Γ un ensemble de formule qui est clos par sou formule. Si $\mathfrak{M}^\star$ est une filtration de $\mathfrak{M}$ par rapport à Γ alors $\mathfrak{M}, m \Vdash A$ si et seulement si $\mathfrak{M}^\star, [m] \Vdash A$ pour tout monde $m \in \mathcal{M}$ et pour toute formule $A \in \Gamma$.*

Démonstration. Les deux sens de la preuve de ce théorème se font par induction sur la complexité de A. Nous détaillons uniquement certains cas et laissons au lecteur la tâche de compléter la preuve pour le cas restant.

La base de l'induction est lorsque A est une formule atomique p.

($\Rightarrow$) Supposons que $\mathfrak{M}, m \Vdash p$, et que $p \in \Gamma$. Par définition de forcing, ceci veut dire que $m \in V(p)$ et donc, par la définition de $L^\star$, que $[m] \in V^\star([m])$. Ceci implique que $\mathfrak{M}^\star, [m] \Vdash p$.

($\Leftarrow$) Supposons que $\mathfrak{M}^\star, [m] \Vdash p$ pour $p \in \Gamma$. par définition de forcing, ceci veut dire que $[m] \in V^\star(p)$. Par définition de $V^\star$ on obtient qu'il existe m' tel que $m \in V(p)$ et $m \equiv_\Gamma m'$. Or $m' \in V(p)$ implique que $\mathfrak{M}, m' \Vdash p$, du moment que $p \in \Gamma$ et $m \equiv_\Gamma m'$ on peut conclure que $\mathfrak{M}, m \Vdash p$.

Pour le pas d'induction, supposons que le théorème est vrai pour toute formule de Γ ayant une complexité qui est inférieure ou égale à n et considérons une formule A appartenant à Γ ayant complexité $n+1$. On traite uniquement les cas $A = \neg B$ et $A = \Diamond B$

1. $A = \neg B$. Supposons que $\mathfrak{M}, m \Vdash \neg B$. Par définition de forcing, ceci veut dire que $\mathfrak{M}, m \nVdash B$. Du moment que B et une de sous formules de $\neg B$ et que Γ est clos par sou formule, on peut appliquer l'hypothèse d'induction et conclure que $\mathfrak{M}^\star, [m] \nVdash B$ et donc que $\mathfrak{M}^\star, [m] \Vdash \neg B$.

   La preuve du sens droite-gauche est complètement symétrique et nous l'omettons.

2. $A = \Diamond B$. Supposons que $\mathfrak{M}, m \Vdash \Diamond B$. Par définition de forcing, on obtient qu'il existe un monde m' tel que $m \mathrel{R} m'$ et $\mathfrak{M}, m' \Vdash B$. Du moment que Γ est clos par sou formules, on applique l'hypothèse d'induction et on conclue que $\mathfrak{M}^\star, [m'] \Vdash B$; Par la condition (2a) dans la définition de Filtration, on obtient que $[m] \mathrel{R^\star} [m']$ et donc on conclue que $\mathfrak{M}^\star, [m] \Vdash \Diamond B$.

   Pour l'autre sens de la preuve : supposons que $\mathfrak{M}, [m] \models \Diamond B$. En dépliant la définition de forcing, on obtient qu'il existe $[m'] \in \mathcal{M}$ tel que $\mathfrak{M}^\star[m'] \Vdash B$ et $[m] \mathrel{R^\star} [m']$. Du moment que B est des sous formule de $\Diamond B$, nous pouvons appliquer l'hypothèse d'induction et conclure que $\mathfrak{M}, m' \Vdash B$. Par la condition (2c) dans la définition de filtration, on obtient que $\mathfrak{M}, m \models \Diamond B$ comme nous le souhaitions.

□

4.3.1 Existence des filtrations

Dans la section précédente, nous avons définis les conditions qu'un modèle de Kripke doit respecter pour être une filtration d'un autre modèle de Kripke par rapport à un ensemble de formules. Malgré cela, nous n'avons pas démontré que la filtration d'un modèle existe toujours. Nous allons donc maintenant démontrer que la filtration d'un modèle existe toujours et que, de plus, ils existent plusieurs filtrations possibles d'un modèle par rapport à un ensemble de formules. En particulier, nous allons nous concentrer sur deux types particuliers de filtrations d'un modèle : la filtration la plus fine (finest filtration en anglais) et la filtration la plus grossière (coarsest filtration en anglais). Du moment que l'ensemble d'états et la fonction d'evaluation d'une filtration

sont fixé par définition (voir la définition 14), la filtration la plus fine et la filtration la plus grossière diffèrent en ce qui concerne leur relation d'accessibilité. La filtration la plus fine aura le moins de mondes apparentés possible, tandis que la filtration la plus grossière en aura autant que possible.

Définition 15. *Soit $\mathfrak{M}$ un modèle et Γ un ensemble de formules qui est clos par sou formule. La* plus fine filtration *$\mathfrak{M}^\star = \langle \mathcal{M}^\star, \mathrm{R}^\star, V^\star \rangle$ de $\mathfrak{M}$ est le modèle de Kripke où :*

1. $\mathcal{M}^\star = \{[m] \mid m \in \mathcal{M}\}$;
2. $V^\star(p) = \{[m] \mid m \in V(p)\}$;
3. $[m]\mathrm{R}^\star[n]$ *si et seulement s'il existe $m' \in [m]$ et il existe $\exists n' \in [n]$ tel que $m' R n'$.*

Proposition 30. *La plus fine filtration $\mathfrak{M}^\star$ d'une modèle $\mathfrak{M}$ par rapport à un ensemble Γ est en effet une filtration.*

Démonstration. Pour établir cette proposition, nous devons vérifier que les conditions (2a), (2b) et (2c) de la définition 14 sont respectées par $\mathfrak{M}^\star$. La condition (2a) est automatiquement remplie : pour tout u $[u]$ est une classe d'équivalence. Nous obtenons donc que $u \in [u]$; si nous supposons que $m\,\mathrm{R}\,n$ on déduit que $[m]\,\mathrm{R}^\star\,[n]$. Pour (2b) supposons que $[m]\,\mathrm{R}\,[n]$, $\Box A \in \Gamma$ et $\mathfrak{M}, m \Vdash \Box A$. Par définition, ceci veut dire qu'ils existent m' et n', tel que $m' \in [m]$, $n' \in [n]$ et $m'\,\mathrm{R}\,n'$. Du moment que $m' \in [m]$ et $\Box A \in \Gamma$, nous déduisons que $\mathfrak{M}, m' \Vdash \Box A$ et vu que $m'\,\mathrm{R}\,n'$, nous pouvons conclure que $\mathfrak{M}, n' \models A$. Puisque Γ est clos par sou formule, nous obtenons que $A \in \Gamma$ et donc, puisque $n' \in [n]$, nous pouvons conclure que $\mathfrak{M}, n' \Vdash A$.

Pour vérifier (2c), supposons que $[m]\,\mathrm{R}^\star\,[n]$, $\Diamond A \in \Gamma$ et $\mathfrak{M}, m \Vdash A$. Par définition de $\mathfrak{M}^\star$, nous obtenons qu'ils existent deux mondes $m' \in [m]$ et $n' \in [n]$ tel que $m'\,\mathrm{R}\,n'$. Vu que $\Diamond A \in \Gamma$, que Γ est clos par sou formule et que $n' \equiv_\Gamma n$, nous déduisons que $\mathfrak{M}, n' \Vdash A$ et comme $m'\,\mathrm{R}\,n'$, nous obtenons que $\mathfrak{M}, m' \Vdash A$. Du moment que $m' \equiv m$, nous pouvons conclure que $\mathfrak{M}, m \Vdash A$. □

Définition 16. *Soit $\mathfrak{M}$ un modèle et Γ un ensemble de formules qui est clos par sou formule. La* plus grossière filtration *$\mathfrak{M}^\star = \langle \mathcal{M}^\star, \mathrm{R}^\star, V^\star \rangle$ de $\mathfrak{M}$ est le modèle de Kripke où :*

1. $\mathcal{M}^\star = \{[m] \mid m \in \mathcal{M}\}$;
2. $V^\star(p) = \{[m] \mid m \in V(p)\}$;
3. $[m]\mathrm{R}^\star[n]$ *si et seulement si les deux conditions qui suivent sont satisfaites :*
   (a) *si $\Box A \in \Gamma$ et $\mathfrak{M}, m \Vdash \Box A$ alors $\mathfrak{M}, n \Vdash A$;*
   (b) *si $\Diamond A \in \Gamma$ et $\mathfrak{M}, n \Vdash A$ alors $\mathfrak{M}, m \Vdash \Diamond A$.*

Nous invitons le lecteur à prouver la proposition suivante.

Proposition 31. *La plus grossière filtration $\mathfrak{M}^\star$ d'une modèle $\mathfrak{M}$ par rapport à un ensemble Γ est en effet une filtration.*

Proposition 32. *Si Γ est un ensemble* fini *de formules qui est clos par sou formule et $\mathfrak{M}^\star$ est une filtration de $\mathfrak{M}$ par rapport à Γ, alors $\mathfrak{M}^\star$ est fini.*

Démonstration. La cardinalité de $\mathcal{M}^\star$ est égal au nombre de classes d'équivalence $[m]$ pour la relation $\equiv_\Gamma$. Considérons la fonction $f : \mathcal{M}^\star \to 2^\Gamma$ qui associe a chaque $[m]$ l'ensemble $[m]_\Gamma$ de formules de Γ qui sont satisfaites par tous les éléments de $[m]$. Il est facile de vérifier que cette fonction est injective : elle associe a deux classes d'équivalence distinctes deux sous-ensembles différentes de formules de Γ. Nous avons donc obtenu qu'il existe une fonction injective qui va de $\mathcal{M}^\star$ aux parties de Γ. Du moment que nous supposons que Γ est fini, son ensemble des parties l'est aussi et nous pouvons conclure. □

Deuxième partie

Quelques familles remarquables de logiques modales

Chapitre 5

Logiques épistémiques

Les logiques épistémiques décrivent avec $\Box$ et $\Diamond$ la connaissance qu'a un agent.

5.1 La logique épistémique $S4$

On peut discuter des axiomes qui modélisent la connaissance d'un agent, disons a, avec pour interprétation de $\Box G$ (dont ce déduit l'interprétation de $\Diamond A = \neg\Box\neg A$:

$\Box A$ noté aussi Ka "l'agent a sait que A"
$\Diamond A$ noté aussi P "A est compatible avec ce que sait l'agent."

S'il y a plusieurs agents a_i on écrit $\Box_a G$ ou $\boxed{a_i} G$ ou encore K_{a_i} avec $(a_i)_{i\in I}$ l'ensemble des agents.

Puisque nous avons affaire à une connaissance rationnelle et non à une croyance, l'axiome T qui correspond à la réflexivité du cadre de Kripke est raisonnable :

$$(T) : \Box A \to A$$

"Si dans une situation donnée l'agent sait que A
alors A est vrai dans cette situation."

L'axiome (4), qui correspond au cadres transitifs est aussi raisonnable pur la connaissance rationnelle, puisqu'il signifie que l'agent, lorsqu'il sait que A a conscience de cette connaissance :

$$(4) : \Box A \to \Box\Box A$$

"Si dans une situation donnée un agent a sait que A, alors l'agent a sait qu'il sait que A."

Pour que la logique modale obtenue puisse s'interpréter dans les modèles de Kripke, il faut bien sur avoir les axiomes de la logique classqiue ainsi que l'axiome κ de Kripke :

$$(\kappa) : \Box(A \to B) \to \Box A \to \Box B$$

"Si dans une situation donnée l'agent sait que $A \to B$ et sait que A alors ils sait que B"

Ce dernier axiome, même s'il est requis pour que cette logique modale de la connaissance fonctionne bien, est discutable. En effet, il signifie que l'agent est logiquement omniscient. Dès qu'il connaît les prémisses du *modus ponens*, il en connaît la conclusion. Cela revient à dire qu'il ses connaissances sont closes par déduction. Ne serait ce que pour des raisons combinatoires, il est clair que c'est une idéalisation de la réalité — à moins que cet agent ne soit un ordinateur ! Lorsque les possibilités de connaissances de l'agent sont finies et en nombres raisonnable, même pour un être humain, l'omniscience logique (κ) est un principe raisonnable.

Comme nous le verrons dans le chapitre sur la logique intuitionniste, très proche de S4, cette logique S4 peut se voir comme la logique d'une connaissance qui progresse, comme la connaissance mathématique. Si on sait que A quelle que soit l'évolution de notre savoir, nous saurons toujours que A.

Définition 17 (S4)**.** *La logique modale dotée des axiome* $(\kappa), (T), (4)$ *s'appelle la logique épistémique S4. Ses modèles sont les cadres réflexifs et transitifs, pour lesquels le théorème de complétude s'applique sous la forme suivante : une formule H est démontrable à partir des axiomes* $(\kappa), (T), (4)$ *(en plus de ceux de la logique classique) si et seulement si elle est valide en tout monde de tout cadre réflexif et transitif. (cf. les exercices 11 et 13 de la section 2.8).)*

5.2 La logique épistémique S5

La logique S5 s'obtient en ajoutant à S4 le principe d'introspection négative. Quel est-il ?

$$(5) : \neg\Box A \to \Box\neg\Box A$$

"si l'agent ne sait pas que A alors il sait qu'il ne sait pas que A"

Si ce principe semble clair lorsqu'il n'y a qu'un nombre fini de chose à savoir comme dans le puzzle des enfants sales (*muddy children*). Sinon il est plus discutable notamment s'ils s'agit d'un agent humain.

Définition 18 (S5)**.** *La logique modale dotée des axiome* $(\kappa), (T), (4), (5)$ *s'appelle la logique épistémique S4. Ses modèles sont les cadres réflexifs, symétriques et transitifs, pour lesquels le théorème de complétude s'applique sous la forme suivante : une formule H est démontrable à partir des axiomes* $(\kappa), (T), (4)$ *(en plus de ceux de la logique classique) si et seulement si elle est valide en tout monde de tout modèle réflexifs et transitifs.*

Remarque 33. *L'accessibilité dans un cadre de S5 est une relation d'équivalence. Deux mondes sont équivalents lorsque l'agent ne peut les distinguer.*

S'il y a plusieurs agents, il y a une relation R^k *d'accessibilité par agent k, et chaque relation d'accessibilité* R^k *est une relation d'équivalence.*

5.2.1 Le puzzle des enfants sales

Ce problème des enfants sales (en anglais *muddy children*) bien connu montre la pertinence de la logique épistémique S5 pour raisonner en prenant en compte les connaissances de chaque agent.

Un nombre n d'enfants ont joué dehors, et certains peuvent avoir de la boue sur le front. Chaque enfant peut voir si les autres enfants ont de la boue sur le front, mais un enfant ne peut voir son propre front, et ne sait donc pas 'il est salle. Le père arrive et leur dit. Au moins un d'entre vous a de la boue sur le front. Puis il leur pose la question

(Q) Que ceux qui savent lèvent la main.

S'il y a n enfants qu'ils répondent $n-1$ fois non (aucun ne sait), à la n-ième question ils lèvent tous la main (et sont tous sales). Il y a des variantes suivant qu'un ou plusieurs enfants répondent plus rapidement.

La logique modale S5 est très adaptée à modéliser ce problème. En effet, vu le peu de nombre de chose à savoir où à ne pas savoir, on accepte volontiers l'omniscience logique (si un agent sait que G alors il sait qu'il sait que G) ainsi que l'introspection négative (si un agent ne sait pas que H alors il sait qu'il ne sait pas que H).

On liste les mondes possibles, chaque relation R^k relie les mondes indiscernables pour l'agent k — c'est bien évidemment une relation d'équivalence.

On trouvera le corrigé comme une série de tableaux.

Chapitre 6

Logique intuitionniste et logique épistémique S4

6.1 La traduction de Gödel de LJ dans S4

On considère les formules de la logique intuitionniste construites avec l'implication inuitionniste notée *impi* et la conjonction intuitionniste notée $\&^j$ (pour les distinguer de l'implication et de la conjonction de S4, notée $\rightarrow$ et &.

On se place dans un système déductif à la Hilbert comme défini dans la section 3.1 au début du chapitre 3.

Rappelons les schémas d'axiomes de la logique intuitionniste dans une présentation à la Hilbert, en ne considérant que l'implication (notée $\rightarrow^j$) et la conjonction (notée $\&^j$:

S : $(p \rightarrow (q \rightarrow^j r)) \rightarrow^j (p \rightarrow^j q) \rightarrow^j (p \rightarrow^j r)$

K : $p \rightarrow^j (q \rightarrow^j p)$

$\&^j$: $p \rightarrow^j (q \rightarrow^j (p \&^j q))$

π_1 : $(p \&^j q) \rightarrow^j p$

π_2 : $(p \&^j q) \rightarrow^j q$

L'unique règle de déduction pour la logique intuitionniste est le modus ponens.
La *traduction de Gödel* est une fonction des formules intuitionnistes vers les formules modales, qui sont écrites avec & et $\rightarrow$. Elle est définie par :

$$\begin{array}{ll} X^{\mathfrak{g}} & = X \\ (A \rightarrow^j B)^{\mathfrak{g}} & = (\Box A^{\mathfrak{g}}) \rightarrow B^{\mathfrak{g}} \\ (A \&^j B)^{\mathfrak{g}} & = A^{\mathfrak{g}} \& B^{\mathfrak{g}} \end{array}$$

Proposition 34. *Pour toute formule intuitionniste A, si A est un théorème intuitionniste (c'est-à-dire si* $A \in \mathsf{IPC}$*) alors la traduction de Gödel de A est valide dans la colletions des cadres qui sont transitifs et réflexifs.*

Démonstration. Il faut montrer que la traduction des formules (1–5) est valide dans tout modèle basé sur un cadre transitif et réflexive. Détaillons les formules $F = X \Rightarrow (Y \Rightarrow X)$ et $G = (X \Rightarrow (Y \Rightarrow U)) \Rightarrow ((X \Rightarrow Y) \Rightarrow (X \Rightarrow U))$

On obtient que

$$F^{\mathfrak{g}} = (\Box X) \to ((\Box Y) \to X)$$

$$G^{\mathfrak{g}} = \Box((\Box X) \to (\Box Y \to U)) \to ((\Box(\Box X \to Y)) \to (\Box X \to U))$$

soit $C = \langle \mathcal{M}, R \rangle$ un cadre dont R est une relation transitive et réflexive, soit $\langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$ un modèle basé sur C et $\mathfrak{m}_i \in \mathcal{M}$. Montrons que $\mathfrak{m}_i \Vdash F^{\mathfrak{g}}$. Supposons que $\mathfrak{m}_i \Vdash \Box Y$ ceci veut dire que, pour tout $j \in I$ si $\mathfrak{m}_i \mathrm{R} \mathfrak{m}_j$ alors $\mathfrak{m}_j \Vdash X$. En particulier on obtient, par réflexivité de R, que $\mathfrak{m}_i \Vdash X$. On a donc deux cas :

1. si $\mathfrak{m}_i \nVdash \Box Y$ on obtient que $\mathfrak{m}_i \Vdash \Box Y \to X$, par définition de l'implication classique.
2. si $\mathfrak{m}_i \Vdash \Box Y$, vu que $\mathfrak{m}_i \Vdash X$, on obtient que $\mathfrak{m}_i \Vdash \Box Y \to X$.

Ensuite, il faut montrer la validité des règles de modus ponens et substitution uniforme. On détaille les modus ponens

Supposons que $(\Box A^{\mathfrak{g}}) \to B^{\mathfrak{g}}$ et $A^{\mathfrak{g}}$ soient valides dans toute modèle base sur un cadre réflexive et transitive. Ceci veut dire que si $\mathfrak{M} = \langle \mathcal{M}, \mathrm{R}, \Vdash_0 \rangle$ est un tel modèle et $\mathfrak{m}_i \in \mathcal{M}$ alors $\mathfrak{m}_i \Vdash A^{\mathfrak{g}}$ et $\mathfrak{m}_i \Vdash (\Box A^{\mathfrak{g}}) \to B^{\mathfrak{g}}$. Du moment que $\mathfrak{M} \models A^{\mathfrak{g}}$ on peut en conclure que $\mathfrak{m}_i \models \Box A^{\mathfrak{g}}$ et donc que $\mathfrak{m}_i \Vdash B^{\mathfrak{g}}$ □

Chapitre 7

Systèmes de transitions

7.1 Un exemple : un distributeur de café et de thé

On se donne un ensemble de propositions atomiques afin de décrire les états de la machine : S qui exprime le fait que la machine est dans l'état initial, E et T qui expriment, respectivement, le fait que la machine est dans un état où l'utilisateur a choisi un expresso ou un thé, $\mathtt{V}_n$ pour $n \in \{0, 10, 20, 30, \ldots\}$ qui exprime le fait qu'à l'état présent la machine a reçu 10,20,30 etc centimes et D-E, D-T qui expriment le fait que la machine donne de l'expresso ou du thé. L'ensemble des actions est les suivantes : deux actions C-E et C-T pour les choix de l'expresso et du thé. Une famille d'actions $\mathtt{I}_m$ où m peut être 2c, 5c, 10c, 20c, 50c, 1 euro ou bien 2 euros. Une famille d'actions $\mathtt{D}_r$ où r est l'un des valeurs précédemment mentionnées.

Exercice 35. *Donner un système de transitions interprété qui modélise la machine à café.*

Exercice 36. *Si on voit les relations du système de transition de l'exercice précédent comme des relations d'accessibilité, chacune d'entre elle désigne une modalité différente. Comment peut-on exprimer les modalités standard "il est possible que ..." et "il est nécessaire que ..." ?*

Exercice 37. *Soit $\mathcal{M}$ le système de transition interprète de l'exercice précédent. Vérifier que les formules suivantes sont valides dans $\mathcal{M}$*

- $S \to \Diamond T \vee \Diamond C$
- $(E \wedge V_0) \to \Diamond (V_0 \vee (\Diamond V_{10} \vee \Diamond V_{20}))$
- $(E \wedge V_{10}) \to \Diamond (V_{10} \vee (\Diamond V_{20} \vee \Diamond V_{30}))$
- $(E \wedge V_{20}) \to \Diamond (V_{20} \vee (\Diamond V_{30} \vee \Diamond V_{40}))$
- $(E \wedge V_{40}) \to \Box V_{30}$
- $(E \wedge V_{30}) \to \Box D\text{-}E$
- $(T \wedge (V_{20} \vee V_{30})) \to \neg \Diamond V_{40}$
- $T \to \neg \Diamond S$
- $D\text{-}E \vee D\text{-}T \to \Box S$

7.2 Introduction

Depuis l'aube de la pensée (occidentale et orientale), les philosophes, théologiens et linguistes utilisent la logique pour raisonner sur le monde, la liberté de choix et la signification du langage humaine. Au milieu du vingtième siècle, la recherche sur les fondements de cette discipline a conduit à deux percées complémentaires. À la fin des années 1950, Arthur Prior a introduit ce que nous appelons aujourd'hui la tense logic ; essentiellement, Prior a introduit les opérateurs $\mathsf{F}\varphi$ signifiant "*ce sera le cas que* φ" et son dual $\mathsf{G}\varphi$ signifiant "*ce sera toujours le cas que* φ. Prior travaillait dans le contexte des logiques modales et son approche était syntaxique. La signification des opérateurs F et G était spécifiée par le philosophe à l'aide d'axiomes en style d' Hilbert et ces opérateurs étaient vus comme des instances de la possibilité $\Diamond$ et nécessite $\Box$.

A peu près à la même époque, le philosophe et mathématicien Saul Kripke, a introduit une sémantique, que nous appelons aujourd'hui sémantique de Kripke, pour interpréter la logique modale. Sa suggestion était d'interpréter cette (famille de) logique(s) sur un ensemble de mondes possibles et une relation d'accessibilité entre les mondes. Les opérateurs modaux sont alors interprétés sur la relation d'accessibilité des mondes. Vingts ans plus tard, ces idées ont pénétré la communauté de la vérification formelle des programmes. Dans un article de référence, Amir Pnueli (prix Turing en 1996 [1]) a montré comment une variante de la logique développé par Prior peut être utilisé pour vérifier la correction des programmes. En particulier, les idées de Pnueli ont pris en compte les calculs continus et non terminaux des programmes. Le paradigme existant de la vérification était celui des programmes de correspondance des pré et post-conditions qui reçoivent une entrée et produisent une sortie à la fin. Au lieu de cela, ce que l'on a appelé plus tard les systèmes réactifs interagissent continuellement avec leur environnement, reçoivent des entrées, envoient des sorties, et surtout ne se terminent pas. Afin de décrire les comportements de tels programmes, il est nécessaire de décrire, par exemple, la causalité des interactions et leur ordre. La logique temporelle s'est avérée un moyen pratique de le faire. Elle peut être utilisée pour capturer la spécification des programmes dans une notation formelle qui peut ensuite être vérifiée sur les programmes.

Systèmes Réactifs Qu'est-ce qu'un programme réactif ? Le schéma général d'exécution d'un programme conventionnel est le suivant : il accepte une entrée, effectue un calcul et produit une sortie. Ainsi, un tel programme peut être considéré comme une fonction abstraite allant d'un domaine d'entrée à un domaine de sortie et dont le comportement consiste en une transformation d'états initiaux en états finaux. Par contre, un programme réactif n'est pas censé se terminer. Comme leur nom l'indique, ces systèmes "réagissent" à leur environnement de manière continue, en répondant de manière appropriée à chaque entrée. Les systèmes d'exploitation, les scheduler, les contrôleurs d'événements discrets, etc. sont des exemples de tels systèmes.

1. Amir Pnueli a reçu le prix Turing en 1996 pour son "travail fondateur introduisant la logique temporelle en informatique et pour des contributions exceptionnelles à la vérification des programmes et systèmes".

7.3 Syntaxe et Sémantique de la Logique Linéaire Temporelle

Avant de présenter la syntaxe et la sémantique de la Logique Linéaire Temporelle (abregé en LTL), nous fixons la terminologie et la notation qui va être utilisé dans la suite.

Notation et Términologie Dans la suite, nous dénotons par $\mathbb{N}$ l'ensemble des nombres naturels et nous considérons que 0 n'est fait pas partie de cet ensemble. Si X est un ensemble, nous dénotons par 2^X l'ensemble des parties de X, i.e., l'ensemble de tous ses sous-ensembles. Si Σ est un ensemble, une *mot* sur Σ est une séquence fini ou infini d'éléments de Σ. Nous dénotons par Σ^+ l'ensemble des séquences fini sur Σ et par Σ^ω l'ensemble des séquences infini sur Σ.

Si ρ est une telle séquence on désigne par ρ_i son i-ème élément, par $\rho_{\leqslant i}$ le préfixe $\rho_1, \ldots, \rho_i$ de ρ et par $\rho_{\geqslant i}$ le suffixe $\rho_i, \rho_{i+1}, \ldots$ de ρ.

7.3.1 Syntaxe

Soit Ap un ensemble au plus dénombrable de formules atomiques. Les formules de la logique linéaire temporelle (LTL) sont définies par la grammaire suivante

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \mathsf{F}\varphi \mid \mathsf{G}\varphi \mid \varphi \mathsf{U} \varphi$$

où p est une formule atomique quelconque. Dans la suite on va utiliser les lettres grecques φ, ψ, et θ (éventuellement indexée par des nombres naturels) pour dénoter des formules quelconques et les lettres $p, q, r \ldots$ etc. pour denoter des formules atomique quelconques.

Les opérateurs unaires X, F, et G sont appelés respectivement **next**, **eventually** et **globaly** et parfois noté $\circ$, $\Diamond$ et $\Box$. On ne va pas utiliser cette notation afin d'éviter toute ambiguïté avec les opérateurs modaux "standard". L'opérateur binaire U est appelé **until**.

7.3.2 Sémantique

Une *interprétation* (ou modèle) linéaire est une fonction $\pi : \mathbb{N} \to 2^{Ap}$, autrement dit un mot ou séquence infini $\pi = \pi_1, \pi_2, \ldots$ d'ensembles de formules atomiques.

Soit π une interprétation linéaire, $i \in \mathbb{N}$ un nombre naturel et φ une formule LTL. Nous écrivons $\pi, i \models \varphi$ pour dénoter que φ est vraie à la position i dans le modèle π. Cette notion est définie par induction sur la structure de φ comme il suit :

— $\pi, i \models \top$ pour toute position i ;
— $\pi, i \models p$ ssi $p \in \pi_i$ pour toute formule atomique p ;
— $\pi, i \models \varphi \wedge \psi$ ssi $\pi, i \models \varphi$ et $\pi, i \models \psi$;
— $\pi, i \models \neg\varphi$ ssi c'est n'est pas le cas que $\pi, i \models \varphi$ (dénoté par $\pi, i \not\models \varphi$) ;
— $\pi, i \models \mathsf{X}\varphi$ ssi $\pi, i+1 \models \varphi$;
— $\pi, i \models \mathsf{F}\varphi$ ssi il existe $j \geqslant i$ tel que $\pi, j \models \varphi$;
— $\pi, i \models \mathsf{G}\varphi$ ssi pour tout $j \geqslant i$ on a que $\pi, j \models \varphi$;

— $\pi, i \models \varphi \mathsf{U} \psi$ ssi il existe $k \geqslant i$ tel que $\pi, k \models \psi$ et $\pi, j \models \varphi$ pour tout $i \leqslant j < k$.

Nous disons qu'une formule φ est **vraie** dans un modèle linéaire π lorsque $\pi_1 \models \varphi$. Nous dénotons cette notion par $\pi \models \varphi$. L'ensemble des modèles linéaires dont la formule φ est vraie est dénoté par $Mod(\varphi)$.

Une formule φ est **valide** lorsqu'elle est vraie dans tout modèle. φ. Deux formules φ et ψ sont équivalents (dénoté par $\varphi \equiv \psi$) lorsque pour toute interprétation π on a que $\pi \models \varphi$ ssi $\pi \models \psi$.

On peut définir les opérateurs booléens habituels $\vee$ et $\rightarrow$ de manière standard. On définit les opérateurs binaires **release** dénoté par R et **weak until** dénote par W de la manière suivante :

$$\varphi \mathsf{R} \psi = \neg(\neg\varphi \mathsf{U} \neg\psi) \tag{7.1}$$

$$\varphi \mathsf{W} \psi = (\varphi \mathsf{U} \psi) \vee \mathsf{G}\varphi \tag{7.2}$$

Exercice 38. *Donner une définition explicite de* $\pi, i \models \varphi \star \psi$ *où* $\star \in \{\mathsf{R}, \mathsf{W}\}$.

La logique LTL nous permet d'exprimer aisément des propriétés des programmes réactives. On rappelle qu'un programme réactif est un programme qui réagit continuellement à des inputs d'un environnement. Par exemple, imaginons qu'on veut exprimer la propriété suivante :

> à chaque instant de l'exécution, si le système détecte une erreur, alors un message signalant l'erreur est affiché à l'écran à l'instant immédiatement successif.

Nous pouvons exprimer cette propriété avec la formule LTL suivante :

$$\mathsf{G}(\mathsf{DetectErreur} \rightarrow \mathsf{X}\,\mathsf{SignalErreur})$$

Nous pouvons compliquer la propriété en demandant que :

> à chaque instant de l'exécution, si le système détecte une erreur, alors à partir de l'instant successif et jusqu'à quand l'erreur n'est plus détectée, un message est affiché à l'écran.

que nous pouvons spécifier à l'aide da la formule qui suit :

$$\mathsf{G}(\mathsf{DetectErreur} \rightarrow \mathsf{X}(\mathsf{SignalErreur} \wedge \mathsf{SignalErreur}\,\mathsf{U} \neg \mathsf{DetectErreur}))$$

Imaginons que l'on souhaite qu'un programme vérifie un nombre infini de fois une certaine propriété φ. Ceci peut être dit en LTL avec la formule :

$$\mathsf{GF}\varphi$$

D'autre part, on peut imaginer de vouloir qu'à partir d'un certain moment de l'exécution du programme, un certaine propriété φ soit toujours vraie ; en utilisant LTL ceci s'exprime par :

$$\mathsf{FG}\varphi$$

7.4 LTL et modèles de Kripke

L'un des moyens plus populaires par lequel on peut représenter l'exécution d'un système réactif sont les systèmes de transitions labellisés. Un système de transition labellisé n'est rien d'autre qu'un graphe orienté dont les nœuds sont étiquetés par des éléments d'un certain alphabet, les arcs sont étiquetés par des éléments d'un certain alphabet et on fixe un ensemble de nœuds qui sont initiaux. En effet, une étape d'exécution d'un programme peut être identifiée par une transition d'un certain ensemble de préconditions à un certain ensemble de postconditions. Une exécution entière peut être identifiée par un flux infini de propriété dont chaque propriété à l'instant t est une précondition de la propriété à l'instant $t+1$.

En vue de la description ci-dessous, on peut définir un système de transition labellisé comme il suit.

Définition 19. *Un système de transition labellisé $\mathcal{T}$ est la donnée de*

- *Deux ensemble non vide et disjoint* Ap *et* Σ*. On considère souvent que* Σ *est fini.*
- *Un ensemble non-vide S d'états et un ensemble non-vide $S_I \subseteq S$ d'états initiaux.*
- *Pour tout $a \in \Sigma$ une relation binaire et sérielle $R_a \subseteq S \times S$ appelé* relation de transition*. Ceci peut être aussi vue comme une fonction binaire qui associe à chaque état s et chaque membre $a \in \Sigma$ un sous-ensemble d'états (éventuellement vide).*
- *une fonction $\mathcal{L} : S \to 2^{\mathsf{Ap}}$ qui associe a chaque état un sous-ensemble (éventuellement vide) de* Ap

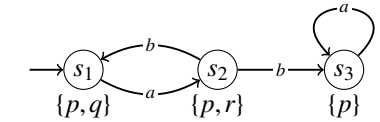
Ainsi, comme le lecteur l'aura compris, un système de transition labellisé n'est rien d'autre qu'une structure de Kripke dont on a choisi un certain nombre de mondes comme étant initiaux, dont la relation de transition est sérielle et dont tout membre de cette relation est nommé par un certain label.

Soit $\mathcal{T} = \langle \mathsf{Ap}, \Sigma, S, s_i, \{R_a\}_{a\in\Sigma}, \mathcal{L}\rangle$ un système de transition. Un chemin de $\mathcal{T}$ est une séquence infini $\rho = s_1, s_2, \ldots$ d'états de S telle que s_1 est un des états initiaux ($s_1 \in S_i$) et pour tout nombre naturel i, on a que $(s_i, s_{i+1}) \in R_a$ pour un certain $a \in \Sigma$. Une **exécution** sur $\mathcal{T}$ est le mot infini sur 2^{Ap} qui résulte de l'application de la fonction $\mathcal{L}$ sur un chemin de $\mathcal{T}$. Plus formellement, si $\rho = s_1, s_2, \ldots$ est un chemin de $\mathcal{T}$ alors $\mathcal{L}(\rho) = \mathcal{L}(s_1), \mathcal{L}(s_2), \ldots$ est une exécution sûr $\mathcal{T}$. On denote par $Ex(\mathcal{T})$ l'ensemble des exécutions sur $\mathcal{T}$, c'est-à-dire : $Ex(\mathcal{T}) = \{\pi \in (2^{Ap})^\omega \mid \pi = \mathcal{L}(\rho)$ pour un chemin ρ de $\mathcal{T}\}$

Définition 20. *Soit $\mathcal{T}$ un système de transition et φ une formule LTL. On dit que $\mathcal{T}$ satisfait universellement φ (denoté par $\mathcal{T} \models_\forall \varphi$) lorsque $\pi \models \varphi$ pour tout $\pi \in Ex(\mathcal{T})$. En français, $\mathcal{T}$ satisfait universellement φ quand toute exécution sur $\mathcal{T}$ rend vraie φ.*

Exemple 39. *Considérons le systèmes de transition ci-dessous $\mathcal{T}_1$. Par convention, l'état initial du système, est l'état avec une flèche entrante qui n'a pas de source. La valeur de la fonction $\mathcal{L}$ pour chaque état s_i est donné en bas de l'état.*

$\mathcal{T}_1 =$



et les formules $\varphi_1 = \mathsf{GF}(p \wedge q) \rightarrow \mathsf{GF}\neg r$. *Et* $\varphi_2 = \mathsf{GF}(p \wedge q) \rightarrow \mathsf{GF}\neg p$. *Nous avons que* $\mathcal{T}_1 \models_\forall \varphi_1$ *(pourquoi ?) et* $\mathcal{T}_1 \not\models_\forall \varphi_2$. *En effet si on considère le chemin* $\rho = s_1, s_2, s_1, s_2, \ldots$ *on a que pour tout* $i \in \mathbb{N}$ *il existe un* $j \geqslant i$ *tel que* $\mathcal{L}(\rho), i \models p \wedge q$ *mais aucun état dans le chemin verifie* $\neg p$.

Lemme 40. *Pour toute formule* φ *de LTL, les affirmations suivantes sont équivalentes :*

1. φ *est valide ;*
2. φ *est universellement satisfait par tout système de transition* $\mathcal{T}$ *;*
3. *l'ensemble des interprétations linéaires dont* $\neg\varphi$ *est vraie est vide.*

Démonstration. La preuve est laissée en exercice au lecteur. □

Si $\mathcal{T}$ est un système de transition, dont l'ensemble des états est S et l'ensemble de proposition atomique est Ap, on dénote par $\mathsf{Ap}(\mathcal{T})$, l'ensemble des éléments de Ap qui apparaissent en tant qu'étiquettes d'un état de $\mathcal{T}$, à savoir

$$\mathsf{Ap}(\mathcal{T}) = \bigcup_{s \in S} \mathcal{L}(s)$$

Si X est un sous-ensemble quelconque de Ap nous dénotons par $Mod_X(\varphi)$ l'ensemble de mots infini sur 2^X qui sont des modèles de φ, c'est-à-dire :

$$Mod_X(\varphi) = \{\pi \in (2^X)^\omega \mid \pi \models \varphi)\}$$

Lemme 41. *Soit* $\mathcal{T} = \langle \mathsf{Ap}, \Sigma, S, s_I, \{R_a\}_{a \in \Sigma}, \mathcal{L}\rangle$ *un système de transition et* φ *une formule LTL sur* Ap. *Les affirmations suivantes sont equivalentes :*

1. $\mathcal{T}$ *satisfait universellement* φ *;*
2. $Ex(\mathcal{T}) \subseteq Mod_{\mathsf{Ap}(\mathcal{T}))}(\varphi)$ *;*
3. $Ex(\mathcal{T}) \cap Mod_{\mathsf{Ap}(\mathcal{T})}(\neg\varphi) = \emptyset$.

Démonstration. La preuve est laissée en exercice au lecteur. □

Satisfiabilité Le problème de la satisfiabilité est le suivante :

> Etant donné une formule LTL φ déterminer s'il existe un intérpretation linéaire π telle que $\pi \models \varphi$.

Validité Le problème de la validité est le suivant

> Étant donné une formule LTL φ déterminer si φ est vraie dans toute interprétation linéaire.

Model-checking Le problème du model-checking est le suivant :

> Étant donné une formule LTL φ et un système de transition **finie** $\mathcal{T}$, déterminer si $\mathcal{T} \models_\forall \varphi$

Le problème de la validité peut être réduit au problème de la satisfiabilité. En effet, φ est validé si et seulement si $\neg\varphi$ n'est pas satisfiable.

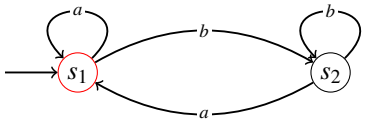
7.5 Automates de Buchi

Un automate fini non déterministe est un tuple $\mathcal{A} = \langle \Sigma, Q, Q_I, \longrightarrow, \alpha \rangle$ dont Σ est un alphabet fini et non vide, Q un ensemble fini et non vide d'états, $Q_I \subseteq Q$ un ensemble d'états initiaux et $\longrightarrow \subseteq Q \times \Sigma \times Q$ une relation de transition et $\alpha \subseteq Q$ un ensemble d'états acceptants. On écrit $q \xrightarrow{a} q'$ lorsque $(q, q', a) \in \longrightarrow$. Notez que l'automate peut être non déterministe, puisqu'il peut avoir plusieurs états initiaux et que la relation de transition peut spécifier plusieurs états successeurs pour chaque lettre a et état q. On dit que l'automate est *déterministe* lorsque $\rightarrow$ est une relation fonctionnelle, c'est-à-dire quand pour tout $q, q', q'' \in Q$ et $a \in \Sigma$ on a que $q \xrightarrow{a} q'$ et $q \xrightarrow{a} q''$ implique $q' = q''$.

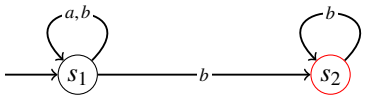
Intuitivement, lorsque l'automate $\mathcal{A}$ s'exécute sur un mot d'entrée sur Σ, il commence dans l'un des états initiaux, et il progresse le long du mot selon la relation de transition.

Formellement, si $\mathbf{a} = a_1, a_2, \ldots$ est un mot infini sur Σ, une *exécution* sur $\mathbf{a}$ est une séquence infini d'états $r = q_1, q_2, \ldots$ telle que $q_1 \in Q_I$ et $q_i \xrightarrow{a_i} q_{i+1}$ pour tout $i \in \mathbb{N}$. Si r est une exécution, on dénote par $Inf(r)$, l'ensemble d'états de Q qui apparaissent une infinité de fois dans r. Formellement $Inf(r) = \{q \in Q \mid q_i = q \text{ pour un nombre indini de i}\}$. Remarquons que $Inf(r)$ est jamais vide, car l'ensemble d'états d'un automate est fini par définition. Comme son nom le suggère, la condition d'acceptation α détermine quelles exécutions sont acceptées par l'automate. La condition d'acceptation de Buchi peut être défini comme il suit : une exécution r est *acceptante* si $Inf(r) \cap \alpha \neq \emptyset$ en mot : une séquence r est acceptante si elle visite une infinité de fois un état acceptante. Étant donné un mot $w \in \Sigma^\omega$, on dit que w est accepté par $\mathcal{A}$ ssi, il existe une exécution r sur w qui est acceptante. Le langage de $\mathcal{A}$ ($\mathcal{L}(\mathcal{A})$) est l'ensemble de mot qui sont acceptés par $\mathcal{A}$.

Exemple 42. *Considérons l'alphabet $\Sigma = \{a, b\}$ et soit $\mathcal{L} \subseteq \Sigma^\omega$ l'ensemble de mots infini w tel que w a un nombre* infini *d'occurrences de la lettre a. Ce langage est reconnu par l'automate de Buchi suivant, ou $\S_I = \alpha = \{s_1\}$. Remarquons que l'automate est déterministe.*



Considérons maintenant le complément $\overline{\mathcal{L}}$ de $\mathcal{L}$, c'est-à-dire l'ensemble de mots infini sur $\{a, b\}$ qui comportent un nombre fini *d'occurrences de a. L'automate suivante (dont l'état acceptante est s_2) reconnait ce langage. Etant donné un mot infini sûr Σ, l'automate "devine" un point de cet input à partir duquel il n'y aura plus d'occurrences de a (un tel point doit exister dans tout mot de $\overline{\mathcal{L}}$). Une fois que l'automaté a choisi ce point, il peut uniquement computer des b, s'il rencontre une occurrence de a il se bloque.*



On peut montrer qu'ils n'existent pas des automates déterministes de Buchi qui sont capables de reconnaître ce dernier langage. Ceci veut dire que, contrairement au cas des automates reconnaissant des mots fini, déterminisme et non-déterminisme ne sont pas équivalents dans le cas des automates de Buchi.

7.5.1 Opérations sur les automates de Buchi

Union Si $\mathcal{A}_1 = \langle \Sigma, Q_1, Q_I^1, \longrightarrow_1, \alpha_1 \rangle$ et $\mathcal{A}_2 = \langle \Sigma, Q_2, Q_I^2, \longrightarrow_2, \alpha_2 \rangle$ sont deux automates de Buchi, on peut construire leur union $\mathcal{A}_1 \cup \mathcal{A}_2$ qui accepté tout le mot sur Σ qui sont acceptés par $\mathcal{A}_1$ et toute celles qui sont acceptés par $\mathcal{A}_2$. On peut supposer que Q_1 et Q_2 sont disjoints et définir $\mathcal{A}_1 \cup \mathcal{A}_2$ par $\langle \Sigma, Q_1 \cup Q_2, Q_I^1 \cup Q_I^2, \longrightarrow_1 \cup \longrightarrow_2, \alpha_1 \cup \alpha_2 \rangle$. Il est clair que chaque mot appartenant au langage de ce dernier automate appartient soit au langage de $\mathcal{A}_1$ ou au langage de $\mathcal{A}_2$.

Intersection Étant donné deux automates de Buchi $\mathcal{A}_1 = \langle \Sigma, Q_1, Q_I^1, \delta_1, \alpha_1 \rangle$ et $\mathcal{A}_2 = \langle \Sigma, Q_2, Q_I^2, \delta_2, \alpha_2 \rangle$ on veut construire une automate de Buchi $\mathcal{A}$ qui reconnaît les mots infinis sur Σ qui sont *à la fois* dans le langage de $\mathcal{A}_1$ et $\mathcal{A}_2$. La façon naturelle d'intersecter les deux automates est de construire un automate dont l'espace d'état est le produit des espaces d'état de $\mathcal{A}_1$ et $\mathcal{A}_2$ et de laisser les deux copies traiter l'entrée simultanément. Pour les mots finis, l'entrée est acceptée si chaque copie peut générer une exécution qui atteint un état final à la fin du mot.

Pour les mots infinis, la situation est plus compliquée et on doit avoir recours à une construction plus sophistiquée. En effet, rien ne nous garantit qu'une computation peut rentrer au meme instant dans un état $q = (q_1, q_2)$ tel que $q_1 \in \alpha_1$ et $q_2 \in \alpha_2$. On doit donc trouver une bonne définition d'états acceptants pour notre automate produit. L'observation clé est que, dans chaque copie, il suffit d'observer une sous-séquence infinie de la séquence globale d'états acceptants. Nous commençons donc par nous concentrer sur la première copie et attendons que sa course entre dans un état acceptant. Lorsque cela se produit, nous portons notre attention sur l'autre copie et attendons qu'elle atteigne un état acceptant. Une fois que la deuxième copie atteint un bon état, nous revenons à la première copie et ainsi de suite. Il est clair que nous passerons de l'un à l'autre infiniment souvent si les deux copies visitent les états acceptants respectifs infiniment souvent. Ainsi, nous pouvons caractériser les états acceptants du produit en termes d'états où l'on passe de l'un à l'autre.

On définit donc $\mathcal{A}$ comme il suit :

— $Q = Q_1 \times Q_2 \times \{1, 2\}$;

— la relation de transition est définie comme il suit :

$$
\begin{array}{llllll}
(q_1, q_2, 1) \xrightarrow{a} (q_1', q_2', 1) & \text{si} & q_1 \xrightarrow{a}_1 q_1' \,,\, q_2 \xrightarrow{a}_2 q_2' & \text{et} & q_1 \notin \alpha_1 \\
(q_1, q_2, 1) \xrightarrow{a} (q_1', q_2', 2) & \text{si} & q_1 \xrightarrow{a}_1 q_1' \,,\, q_2 \xrightarrow{a}_2 q_2' & \text{et} & q_2 \in \alpha_1 \\
(q_1, q_2, 2) \xrightarrow{a} (q_1', q_2', 2) & \text{si} & q_1 \xrightarrow{a}_1 q_1' \,,\, q_2 \xrightarrow{a}_2 q_2' & \text{et} & q_2 \notin \alpha_2 \\
(q_1, q_2, 2) \xrightarrow{a} (q_1', q_2', 1) & \text{si} & q_1 \xrightarrow{a}_1 q_1' \,,\, q_2 \xrightarrow{a}_2 q_2' & \text{et} & q_2 \in \alpha_2
\end{array}
$$

— L'ensemble des états initiaux est $\{(q_1, q_2, 1) \mid q_1 \in Q_I^1, q_2 \in Q_I^2\}$.

— $\alpha = Q_1 \times \alpha_2 \times \{2\}$

Dans l'automate que l'on vient de définir, chaque état du produit porte une étiquette supplémentaire indiquant si l'automate vérifie un état acceptant sur le premier ou le second composant. Par construction, l'automate accepte s'il passe du second composant au premier infiniment souvent. Le lecteur peut vérifier que le langage reconnu par $\mathcal{A}$ est l'intersection entre le langage de $\mathcal{A}_1$ et le langage de $\mathcal{A}_2$.

Test du vide Soit $\mathcal{A}$ un automate de Buchi. On veut savoir si $\mathcal{L}(\mathcal{A}) \neq \emptyset$. Autrement dit, si l'automaté accepté au moins un mot w. Considérons le graphe $\mathcal{G} = (V, E_{\mathcal{A}})$ dont l'ensemble V des nœuds est l'ensemble des états de l'automate et dont deux nœuds s et t sont adjacents (i.e., $(s,t) \in E_{\mathcal{A}}$) si et seulement si $s \xrightarrow{a} t$ pour un certain a dans l'alphabet Σ de l'automate. Il est aise de montrer que le langage de l'automate est vide si et seulement si G ne contient pas un chemin p qui part d'un des états initiaux, rejoins un état acceptant v et puis revient à cet état acceptant. En effet : supposons que dans $\mathcal{G}$ il existe un chemin $p = p_1, \ldots, p_n$ tel que $p_1 \in Q_I$, $p_i \in \alpha$ et $p_i = p_n$. Par définition, ceci veut dire qu'il existe une exécution r sur $\mathcal{A}$ telle que $r = p_1, \ldots p_{i-1} \cdot (p_i, \ldots p_n)^{\omega}$. Donc r contient une infinité de fois un état acceptante. Réciproquement, supposons que le langage de l'automate est non vide. Donc il existe une exécution r telle que $inf(r) \cap \alpha \neq \emptyset$. Du moment que α est fini, il doit exister un état $v \in \alpha$ qui apparait une infinité de fois dans r. Du moment que Q est fini, on obtient que dans le graphe $\mathcal{G}$ il existe un chemin qui arrive v depuis un état initial (car l'état r_1 de l'exécution est initial par définition) et revient à v. On peut utiliser un algorithme de parcours en profondeur sûr $\mathcal{G}$ pour tester s'il existe un chemin ayant la caractéristique qu'on vient de définir. On obtient donc le théorème suivant :

Théorème 43. *Étant donné un automate de Buchi $\mathcal{A}$, on peut décider en temps linéaire si le langage de l'automate est vide.*

7.6 Automates de Buchi, Satisfiabilité et Model-Cheking

TBW

7.7 La logique CTL

7.7.1 Arbre des computations

Nous allons maintenant introduire la logique CTL (Computation Tree Logic). Comme son nome le suggère, CTL est utilisé pour raisonner sur l'arbre des computations d'un programme. Cet arbre aura comme racine l'instant de départ t_0 du programme et comme branche toutes les exécutions possibles du programme qui commence à t_0. Pour définir la syntaxe et la sémantique de CTL nous n'avons pas besoin de définir formellement la notion d'arbre des computations. Néanmoins, nous allons la définir afin de rendre la comprehension de cette logique plus simple.

Avant de formaliser la notion d'arbre des computations, fixons la terminologie.

Un **arbre** est un graphe orienté (fini ou infini) $\mathcal{G} = (V, E)$ qui respecte les deux propriétés suivantes :

1. Il existe un nœud r (appelé racine) qui n'a pas de parentes ; autrement dit : il existe $r \in V$ tel que pour tout $x \in V$ on a que $(x, r) \notin E$.
2. Pour tout nœud x de V si x n'est pas la racine, alors il existe un unique nœud y tel que y est le parent de x, c-a-d $(y, x) \in E$;

Un chemin dans un arbre est une séquence (fini ou infini) de nœuds $x_1, x_2, \ldots$ tels $x_1 = r$ et x_i est le parent de x_{i+1} pour tout nombre naturel i. Si ρ est un chemin fini de l'abre $\rho = x_1, \ldots, x_n$ alors $last(\rho)$ dénote le dernier élément x_n de ρ

Une branche est un chemin qui est maximale par rapport à l'ordre par préfixe. Un arbre labellisé est la donnée d'un arbre et d'une fonction $\mathcal{L}$ qui associé à tout nœud de l'arbre un élément d'un certain ensemble.

Définition 21. *Soit $\mathcal{T} = \langle S, s_I, R, \mathcal{L}\rangle$ un système de transition labellisé. L'arbre des computations de $\mathcal{T}$ est le graphe orienté labellisé $\mathcal{G}_\mathcal{T} = \langle V_\mathcal{T}, E_\mathcal{T}, \mathcal{L}_\mathcal{T}\rangle$ dont :*

1. *l'ensemble des nœuds $V_\mathcal{T}$ est l'ensemble des séquences fini sur S qui commencent à s_I ;*
2. *étant donné deux séquences finies ρ et τ sur S, on a que ρ est un parent de τ (c-à-d $(\rho, \tau) \in E_\mathcal{T}$) si et seulement si $\tau = \rho \cdot s$ pour un certain $s \in S$ et $(last(\rho), s) \in R$;*
3. *$\mathcal{L}_\mathcal{T}(\rho) = \mathcal{L}(last(\rho))$ pour toute séquence finie ρ sur S.*

Proposition 44. *Étant donné un système de transition $\mathcal{T} = \langle S, s_I, R, \mathcal{L}\rangle$ on obtient que le graphe orienté labellisé $\mathcal{G}_\mathcal{T} = \langle V_\mathcal{T}, E_\mathcal{T}, \mathcal{L}_\mathcal{T}\rangle$ est un arbre labellisé.*

Démonstration. La démonstration est laissée en exercice au lecteur.

□

Exercice 45. *Soit $\mathcal{T}$ le système de transition dont $S = \{s_1, s_2\}$, $s_i = s_1$, $R = \{(s_1, s_1), (s_1, s_2), (s_2, s_2)\}$ et $\mathcal{L} = \{(s_1, \{p\}), (s_2, \{q\})\}$. Dessiner $\mathcal{T}$ et dessiner l'arbre de computation de $\mathcal{T}$ (autant que possible).*

7.7.2 Syntaxe et Sémantique

Maintenant que nous avons vu comment déplier un système de transition dans son arbre des computations, nous pouvons définir la syntaxe et la sémantique de la logique CTL. Soit Ap un ensemble au plus dénombrable de proposition atomiques. Les formules de la logique CTL sont définies par la grammaire suivante

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \mathsf{EF}\varphi \mid \mathsf{AF}\varphi \mid \mathsf{EG}\varphi \mid \mathsf{AG}\varphi \mid \mathsf{E}\varphi\mathsf{U}\varphi \mid \mathsf{A}\varphi\mathsf{U}\varphi$$

où $p \in \mathsf{Ap}$ est une proposition atomique. Les connecteurs booléens $\vee$ et $\rightarrow$ se définissent comme d'habitude. On va utiliser φ, ψ et θ pour désigner de formules arbitrer CTL.

Soit $\mathcal{T} = \langle S, s_I, R, \mathcal{L}\rangle$ un système de transition. Comme d'habitude, un chemin (dans $\mathcal{T}$) est une séquence infinie $s_1, s_2, s_3, \ldots$ d'états telle que $(s_i, s_{i+1}) \in R$ pour tout nombre naturel i. Comme d'habitude pour tout $i \in \mathbb{N}$, ρ_i désigne le i-ème élément du chemin. On définit "$\mathcal{T}$ satisfait φ à l'état s" (écrit $\mathcal{T}, s \models \varphi$) par induction sur la structure de φ

— $\mathcal{T}, s \models \top$ pour tout $s \in S$;
— $\mathcal{T}, s \models p$ ssi $p \in \mathcal{L}(s)$;
— $\mathcal{T}, s \models \neg\varphi$ ssi c'est n'est pas le cas que $\mathcal{T}, s \models \varphi$;
— $\mathcal{T}, s \models \varphi \wedge \psi$ ssi $\mathcal{T}, s \models \varphi$ et $\mathcal{T}, s \models \psi$
— $\mathcal{T}, s \models \mathsf{EX}\varphi$ ssi il existe un chemin ρ tel que $\rho_1 = s$ et $\mathcal{T}, \rho_2 \models \varphi$
— $\mathcal{T}, s \models \mathsf{AX}\varphi$ ssi pour tout chemin ρ tel que $\rho_1 = s$ on a que $\mathcal{T}, \rho_2 \models \varphi$
— $\mathcal{T}, s \models \mathsf{EF}\varphi$ ssi, il existe un chemin ρ tel que $\rho_1 = s$ et il existe $j \geqslant 1$ tel que $\mathcal{T}, \rho_j \models \varphi$
— $\mathcal{T}, s \models \mathsf{AF}\varphi$ ssi, pour tout chemin ρ tel que $\rho_1 = s$ il existe $j \geqslant 1$ tel que $\mathcal{T}, \rho_j \models \varphi$
— $\mathcal{T}, s \models \mathsf{EG}\varphi$ ssi il existe un chemin ρ tel que $\rho_1 = s$ et pour tout $j \geqslant 1$ on a que $\mathcal{T}, \rho_j \models \varphi$
— $\mathcal{T}, s \models \mathsf{AG}\varphi$ ssi pour tout chemin ρ tel que $\rho_1 = s$ et pour tout $j \geqslant 1$ on a que $\mathcal{T}, \rho_j \models \varphi$
— $\mathcal{T}, s \models \mathsf{E}\varphi\mathsf{U}\psi$ ssi il existe un chemin ρ tel que $\rho_1 = s$ et il existe $j \geqslant 1$ tel que $T, \rho_j \models \psi$ et $\mathcal{T}, \rho_k \models \varphi$ pour tout $1 \leqslant k < j$ $\mathcal{T}, \rho_j \models \varphi$
— $\mathcal{T}, s \models \mathsf{A}\varphi\mathsf{U}\psi$ ssi pour tout chemin ρ tel que $\rho_1 = s$ il existe $j \geqslant 1$ tel que $T, \rho_j \models \psi$ et $\mathcal{T}, \rho_k \models \varphi$ pour tout $1 \leqslant k < j$

Exercice 46. *Soient* `crash` *et* `finish` *deux formules atomiques exprimant (respectivement) que le système est en panne et que le système a fini de s'exécuter. Donner trois formules CTL φ_1, φ_2 et φ_3 qui traduisent les trois spécifications suivantes.*

1. *Le système peut évoluer de telle sorte qu'il ne tombe jamais en panne.*

2. *Quelle que soit l'évolution du système, il ne va jamais en panne.*

3. *Il existe un moyen de faire planter le système dans deux états consécutifs, mais il n'est pas possible que le système se plante une fois l'exécution terminée.*

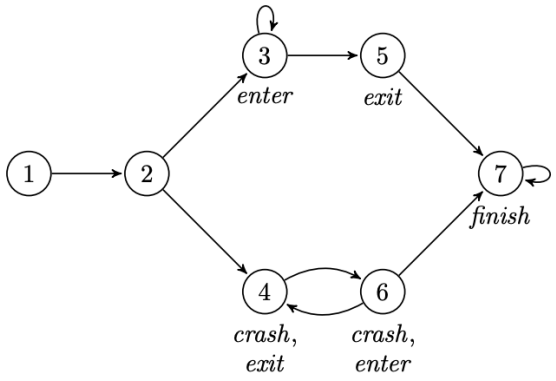
Solution. *Voici une solution possible*

$\varphi_1 = \mathsf{EG}\neg \texttt{crash}$

$\varphi_2 = \mathsf{AG}\neg \texttt{crash}$

$\varphi_3 = (\mathsf{EF}(\texttt{crash} \wedge \mathsf{EX}\,\texttt{crash})) \wedge (\mathsf{AG}\,\texttt{finish} \rightarrow \neg \texttt{crash})$

Exercice 47. *Soit $\mathcal{T}$ le système de transition ci-bas. Déterminer si les formules φ_1, φ_2 et φ_3 sont satisfaites aux états 1,3 et* 4.



Solution. *Voici une correction :*

— *Pour φ_1 le chemin $\rho^1 = 1,2,3,5,7^\omega$ satisfait clairement la spécification. Donc, on obtient que $\mathcal{T},1\varphi_1$. De moment que $\rho^2 = 3,5,7^\omega$ est un suffixe de ρ on obtient aussi que $\mathcal{T},3 \models \varphi_1$. Pour 4 on a que che $\texttt{crash} \in \mathcal{L}(4)$, donc tout chemin qui part à 4 ne peut pas satisfaire $\mathsf{G}\neg\texttt{crash}$. Donc $\mathcal{T},4 \not\models \varphi_1$.*

— *Pour φ_2 ; Le chemin $\tau^1 = 1,2,(4\cdot 6)^\omega$ ne satisfait pas $\mathsf{G}\neg\texttt{crash}$. Donc il existe au moins un chemin qui part à 1 et qui ne satisfait pas $\mathsf{G}\neg\texttt{crash}$ et par conséquent $\mathcal{T},1 \not\models \varphi_2$. Les chemins qui partent à 3 sont : pour tout $n \in \mathbb{N}$, $3^n,5,7^\omega$, et 3^ω. Du moment que $\texttt{crash} \notin \mathcal{L}(j)$ pour $j \in \{3,5,7\}$ on obtient que tout chemin qui part a 3 satisfait $\mathsf{G}\neg\texttt{crash}$. On en conclut que $\mathcal{T},3 \models \varphi_2$. Du moment que $\mathcal{T},4 \not\models \varphi_1$ on peut conclure que $\mathcal{T},4 \not\models \varphi_2$.*

— *Pour φ_3. Remarquons d'abord que chaque état de $\mathcal{T}$ vérifie $\psi = \texttt{finish} \rightarrow \neg\texttt{crash}$, donc $\mathsf{AG}\psi$ est vérifiée partout sur $\mathcal{T}$. Considérons les chemins $\tau = 4,6,7^\omega$ et $\tau' = (4,6)^\omega$ ce deux chemins sont les seuls à vérifier $\mathsf{F}(\texttt{crash} \wedge \mathsf{EX}\,\texttt{crash})$. On obtient donc que $\mathcal{T},4 \models \varphi_3$ et puisque $1,2\cdot\tau$ est un chemin qui part à l'état 1 on a aussi que $\mathcal{T},1 \models \varphi_3$. Du moment qu'on ne peut pas rejoindre le deux chemins τ et τ' en partant de 3 on obtient que $\mathcal{T},3\neg \models \varphi_3$.*

Exercice 48. *Dans cet exercice, on va démontrer que la logique CTL restante aux connecteurs EX, EG et $\mathsf{E\,U}$ est aussi puissante que la logique CTL entière. Plus précisément : on va montrer que pour toute formule CTL φ il existe une formule CTL ψ écrit uniquement avec les connecteurs mentionné ci-haut et telle que $\mathcal{T},s \models \varphi$ ssi $\mathcal{T},s \models \psi$ pour tout $\mathcal{T}$ et s. Pour le faire, on va montrer que :*

1. $\mathsf{EF}\varphi \equiv \mathsf{E}\top\mathsf{U}\varphi$;
2. $\mathsf{AX}\varphi \equiv \neg\mathsf{EX}\neg\varphi$
3. $\mathsf{AG}\varphi \equiv \neg\mathsf{EF}\neg\varphi$
4. $\mathsf{AF}\varphi \equiv \neg\mathsf{EG}\neg\varphi$
5. $\mathsf{A}\varphi\mathsf{U}\psi \equiv \neg(\mathsf{E}(\neg\psi\,\mathsf{U}\,\neg(\phi\vee\psi)) \vee \mathsf{EG}(\neg\psi))$

Solution. *On corrige uniquement 5. Toutes les autres equivalences sont triviales.*

1. *Omise ;*
2. *Omise ;*
3. *Omise ;*
4. *Omise ;*
5. *Supposons que* (i) $\mathcal{T}, s \models \mathsf{A}\varphi\mathsf{U}\psi$ *et (pour obtenir une contradiction) que* $\mathcal{T}, s \models \mathsf{E}(\neg\psi\mathsf{U}\neg(\phi\vee\psi))\vee\mathsf{EG}(\neg\psi)$. *Si* $\mathcal{T}, s \models \mathsf{E}((\neg\psi\mathsf{U}\neg(\phi\vee\psi)))$ *alors il existe un chemin* ρ *qui part à* s *tel que* $\mathcal{T},\rho_k \models \neg\varphi\wedge\neg\psi$ *et* $\mathcal{T},\rho_j \models \neg\psi$ *pour tout* $1 \leqslant j < k$. *Du moment qu'on suppose que* (i) *on doit avoir qu'il existe un* $k' \geqslant k$ *tel que* $\mathcal{T},\rho'_k \models \psi$ *et* $\mathcal{T},\rho'_j \models \varphi$ *pour tout* $1 \leqslant j' > k'$ *mais ceci est impossible pour* $j' = k$. *Contradiction. Si* $\mathcal{T}, s \models \mathsf{EG}\neg\varphi$ *on obtient immédiatement une contradiction par* (i).
   Pour l'autre sens, supposons que $\mathcal{T}, s \models \neg(\mathsf{E}(\neg\psi\mathsf{U}\neg(\phi\vee\psi))\vee\mathsf{EG}(\neg\psi))$. *Par définition, ceci veut dire que* $\mathcal{T}, s \not\models \mathsf{E}(\neg\psi\mathsf{U}\neg(\phi\vee\psi))$ *et* $\mathcal{T}, s \not\models \mathsf{EG}\neg\psi$. *Ceci implique que* $(ii)\mathcal{T}, s \models \mathsf{AF}\psi$. *Du coup, on a obtenu que pour tout* ρ *qui part à* s *il existe* j *tel que* $\mathcal{T},\rho_j \models \psi$. *Maintenant, du moment que* $\mathcal{T}, s \not\models \mathsf{E}(\neg\psi\mathsf{U}\neg(\phi\vee\psi)))$ *on en déduit que sur tout chemin* ρ *qui part à* s, *la formule* $\varphi\vee\psi$ *est vraie tant que* ψ *ne devient vraie. En particulier, ceci implique que si* ψ *n'est pas vraie alors* φ *l'est. Par (ii) on sait que tout chemin qui part à* s *rejoins un état ou* ψ *est vraie. On peut donc conclure.*

7.7.3 Model-Checking pour CTL

Si $\mathcal{T} = \langle S, s_I, R, \mathcal{L}\rangle$ est un système de transition et φ une formule CTL on dénote par $[\![\varphi]\!]^{\mathcal{T}}$ l'ensemble d'états de $\mathcal{T}$ qui satisfont φ, autrement dit $[\![\varphi]\!]^{\mathcal{T}} = \{s \in S \mid \mathcal{T}, s \models \varphi\}$. Nous pouvons énoncer le problème de model-checking comme il suit.

Définition 22. *Étant donné un système de transition* fini $\mathcal{T}$ *et une formule CTL* φ, *le problème du model-checking consiste à déterminer* $[\![\varphi]\!]^{\mathcal{T}}$.

Dans la suite, nous allons montrer qu'il existe un algorithme efficace (linéaire dans la cardinalité de S et R) pour résoudre ce problème.

Pour un état s de $\mathcal{T}$, $pre(s)$ dénote l'ensemble des prédécesseurs de s par la relation R, autrement dit $pre(s) = \{s' \in S \mid (s', s) \in R\}$. Si A est un sous-ensemble de S $Pre(A)$ est $\bigcup_{s\in A} pre(s)$.

Exercice 49. *Soit* φ *une formule CTL et* $\mathcal{T}$ *un système de transition, montrer que* $[\![\mathsf{EX}\varphi]\!]^{\mathcal{T}} = Pre([\![\varphi]\!]^{\mathcal{T}})$

Solution. *Supposons que* $\mathcal{T}, s \models \mathsf{EX}\varphi$. *Par définition il existe un chemin* ρ *qui part à* s *et tel que* $\mathcal{T},\rho_2 \models \varphi$. *Par définition du chemin on a que* $(s,\rho_2) \in R$ *et puisque* φ *est vriaie à* ρ_2 *on obtient que* $s \in Pre(\varphi)$.

Maintenant, supposons que $s \in Pre([\![\varphi]\!])$, *ceci veut dire que il existe* s' *tel que* $(s, s') \in R$ *et* $s' \in [\![\varphi]\!]$. *Soit donc* ρ *un chemin tel que* $\rho_1 = s$ *et* $\rho_2 = s'$. *Ce chemin verifie* $\mathsf{X}\varphi$ *et on obtient donc le résultat.*

Soit X un ensemble et F une fonction de 2^X dans 2^X. Un **point fixe** pour F est une partie Y de X tel que $F(Y) = Y$. On dit que F est monotone si pour tout $Y,Z \subseteq X$ on

obtient que $Y \subseteq Z$ implique, $F(Y) \subseteq F(Z)$. Le théorème de Knaster-Tarsky nous dit que si F est monotone, alors elle admet un plus petit et un plus grand point fixe. C'est-à-dire :

1. il existe un point fixe Y' de F tel que $Y' \subseteq Z$ pour tout point fixe Z
2. il existe un point fixe Y'' de F tel que $Z \subseteq Y''$ pour tout point fixe Z.

Comme on va comprendre, l'existence du point fixe pour une fonction monotone sur les parties d'un ensemble est une propriété essentielle pour la terminaison, correction et complétude de l'algorithme de model-checking pour CTL. En effet, nous allons montrer que pour toutes formules CTL φ et ψ et pour tout système de transition, $[\![\mathsf{EG}\varphi]\!]^{\mathcal{T}}$ et $[\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^{\mathcal{T}}$ sont (respectivement) le plus grand et plus pétit point-fixe d'un certaine fonction monotone.

Exercice 50 (Équivalences de point-fixe)**.** *Soient φ et ψ deux formules CTL quelconques. Montrer que*

1. $\mathsf{EG}\varphi \equiv \varphi \wedge \mathsf{EXEG}\varphi$
2. $\mathsf{E}\varphi\mathsf{U}\psi \equiv \psi \vee (\varphi \wedge \mathsf{EXE}\varphi\mathsf{U}\psi)$

Solution. *Voici une solution pour le premier point. Le deuxième peut se resoudre de manière similaire/*

1. *($\Rightarrow$) Supposons que $\mathcal{T}, s \models \mathsf{EG}\varphi$. Ceci veut dire qu'il existe un chemin ρ qui part à s et tel que $\mathcal{T}, \rho_j \models \varphi$ pout tout nombre naturel j. Ceci veut dire, en particulier, que pour $\rho_1 = s$ on a $\mathcal{T}, s \models \varphi$. On a aussi que pour tout $j' \geqslant 2$, $\mathcal{T}, \rho'_j \models \varphi$ et donc $\mathcal{T}, \rho_1 \models \mathsf{XG}\varphi$ comme on voulait.*
   ($\Leftarrow$) Supposons que $\mathcal{T}, s \models \varphi \wedge \mathsf{EXG}\varphi$. Ceci veut dire qu'il existe un chemin ρ tel que $\rho_1 = S$ et tel que $\mathcal{T}, \rho_2 \models \mathsf{EG}\varphi$. Par la sémantique de CTL on obtient qu'il existe un chemin τ tel que $\tau_1 = \rho_2$ et $\mathcal{T}, \tau_i \models \varphi$ pour tout $i \geqslant 1$. Considerons donc le chemin $\pi = s \cdot \tau$. Ce chemin verifie $\mathsf{G}\varphi$ par définition. On obtient donc que $\mathcal{T}, s \models \mathsf{EG}\varphi$.

Soit $\mathcal{T}$ un système de transition et φ, ψ deux formules CTL. Considérons les fonctions EG_φ et $\mathsf{E}\,\mathsf{U}_{\varphi,\psi}$ définies par

$$\mathsf{EG}(X) = [\![\varphi]\!]^{\mathcal{T}} \cap (Pre(X)) \tag{7.3}$$

$$\mathsf{EG}(X) = [\![\psi]\!]^{\mathcal{T}} \cup ([\![\varphi]\!]^{\mathcal{T}} \cap Pre(X)) \tag{7.4}$$

Par l'exercice précédent, nous obtenons immédiatement le théorème qui suit

Théorème 51. *Soit $\mathcal{T}$ un système de transition et φ et ψ deux formules CTL. Nous avons que :*

1. $[\![\mathsf{EG}\varphi]\!]^{\mathcal{T}}$ *est le plus grand point fixe de la fonction* EG_φ *;*
2. $[\![\mathsf{E}\varphi\mathsf{U}\psi]\!]^{\mathcal{T}}$ *est le plus petit point fixe de la fonction* $\mathsf{EU}_{\varphi,\psi}$ *;*

Démonstration. Nous allons prouver (2) la preuve de (1) est plus simple et elle est laissée en exercice.

D'abords, nous nous convainquons que $X = [\![\mathsf{E}\varphi\mathsf{U}\psi]\!]$ est bel et bien un point fixe de $\mathsf{E}\mathsf{U}_{\varphi,\psi}$. Pour le faire, il est suffisant de remarquer que, par l'exercice 50, nous obtenons que $X = [\![\psi \vee (\varphi \wedge \mathsf{E}\mathsf{X}\mathsf{E}\varphi\mathsf{U}\psi)]\!]^{\mathcal{T}}$. De cette équation, nous obtenons que $X = [\![\psi]\!] \cup ([\![\varphi]\!] \cap [\![\mathsf{E}\mathsf{X}\varphi\mathsf{U}\psi]\!]$. Finalement, par l'exercice 49 nous obtenons que $X = [\![\psi]\!] \cup ([\![\varphi]\!] \cap Pre([\![\mathsf{E}\varphi\mathsf{U}\psi]\!])$ comme on le souhaitait. Pour démontrer que X est le plus petit point fixe de la fonction que l'on considère, on fixe un autre point fixe Y et montre que si $s \in X$ alors $s \in Y$. Par définition de X on a qu'il existe un chemin ρ tel que $\rho_1 = s$ et un nombre naturel $n \geqslant 1$ tel $\rho_n \in [\![\psi]\!]$ et pour tout k tel que $1 \leqslant k < n$, $\rho_k \in [\![\varphi]\!]$. D'abord, nous obtenons que $\rho_n \in [\![\psi]\!]$ et puisque $Y = [\![\psi]\!] \cup ([\![\varphi]\!] \cap Pre(Y))$ on obtient que $\rho_n \in Y$. Nous montrons maintenant que si $\rho_i \in Y$ alors $\rho_{i-1} \in Y$ ce qui implique que $s \in Y$. Supposons que $\rho_i \in Y$. Par définition de chemin, nous obtenons que $(\rho_{i-1}, \rho_i) \in R$ et puisque $\rho_{i-1} \in [\![\varphi]\!]$ on obtienne que $\rho_{i-1} \in [\![\varphi]\!] \cap Pre(Y)$ puisque ce dernier ensemble est inclus dans Y, nous déduisons que $\rho_{i-1} \in Y$ et nous pouvons conclure.

□

7.7.4 L'algorithme de Model-Checking

Pour une formule φ, on dénote par $Sub(\varphi)$ la liste de sous formule ordonnée par ordre de complexité croissant. Voici un algorithme 1 de model-checking pour la logique CTL.

Analysons l'algorithme.

— Si φ est une variable propositionnelle ou une formule booléenne, l'algorithme est linéaire sur les nombres d'états. Le fait que l'algorithme soit correcte est complète dans ces cas de figure est évidant.

— si $\varphi = \mathsf{E}\mathsf{X}\varphi_1$ alors, il faut considérer le cout de calculer la $Pre(X)$ pour un certain ensemble $X \subseteq S$. Si on utilise une matrice d'adjacence pour représenter la relation, R on obtient que chequer si $(s, s') \in R$ se fait en temps constant. $X = S$ est le pire de cas ; dans ce cas de figure, la computation de $Pre(X)$ doit faire $|R|$ fois un calcul constant. On obtient donc que $Pre(X)$ est linéaire sur la taille $|R|$ de la relation de transition.

— Si $\varphi = \mathsf{E}\mathsf{G}\varphi_1$ ou $\varphi = \mathsf{E}\varphi_1\mathsf{U}\varphi_2$ on est garanti que les boucle se terminent en un nombre fini d'étapes (au plus $|S|$). À chaque étape, les deux boucles produisent un ensemble d'état. De plus, le résultat de la boucle à l'étape t est un sous ensemble du résultat de la boucle à l'instant $t+1$. Ceci veut dire qu'ils calculent des chaînes de sous-ensembles de l'ensemble des états, et la longueur maximale de ces chaînes est limitée par la cardinalité de S qui est finie. Enfin, les boucles dans les deux cas peuvent également être exécutées en temps $O(|R|)$ parce qu'elles peuvent être implémenté de manière que chaque transition ne soit prise en compte qu'une seule fois en commençant par φ_2 par exemple et en visitant successivement les prédécesseurs. La correction et la complétude de l'agorithme pour ce deux cas vient du fait qu'ils calculent exactement le deux fonctions des équations 7.3 et 7.4.

Pour conclure, il est clair qu'il faut au plus $|\varphi|$ itérations de la boucle à travers

Algorithm 1 MC Algorithm $(\mathcal{T}, \varphi)$

for all $\varphi \in Sub(\varphi)$ **do**
  switch φ **do**
    case $\varphi = \top$
      $[\![\varphi]\!] \leftarrow S$
    case $\varphi = p$
      $[\![\varphi]\!] \leftarrow \{s \in S \ : \ p \in \mathcal{L}(s)\}$
    case $\varphi = \neg\varphi_1$
      $[\![\varphi]\!] \leftarrow S \setminus [\![\varphi_1]\!]$
    case $\varphi = \varphi_1 \wedge \varphi_2$
      $[\![\varphi]\!] \leftarrow [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$
    case $\varphi = \mathsf{EX}\varphi_1$
      $[\![\varphi]\!] \leftarrow Pre([\![\varphi_1]\!])$
    case $\varphi = \mathsf{EG}\varphi_1$
      $X \leftarrow [\![\top]\!]$; $Y \leftarrow [\![\varphi_1]\!]$
      while $X \neq Y$ **do**
        $X \leftarrow Y$
        $Y \leftarrow [\![\varphi_1]\!] \cap Pre(X)$
      $[\![\varphi]\!] \leftarrow Y$
    case $\varphi = \mathsf{E}(\varphi \mathsf{U}_1 \varphi_2)$
      $X \leftarrow \emptyset$; $Y \leftarrow [\![\varphi_2]\!]$
      while $Y \neq X$ **do**
        $X \leftarrow Y$
        $Y \leftarrow [\![\varphi_2]\!] \ \cup ([\![\varphi_1]\!] \cap Pre(X))$
      $[\![\varphi]\!] \leftarrow Y$

toutes les sous-formules. Nous obtenons donc que :

Théorème 52. *Soit $\mathcal{T} = \langle S, s_i, R, \mathcal{L}\rangle$ un système de transition et φ une formule. L'ensemble d'états qui satisfont φ peut être calculé en temps $O(|R| \cdot |\varphi|)$.*

Troisième partie

Logique modale et premier ordre

Chapitre 8

Traduction logiques modales en logique du premier ordre

8.1 Rappel : langage et interprétation du premier ordre

Dans la suite, nous nous intéressons à des langages du premier ordre restreint sans symboles de fonctions ni symboles de constantes. Voici leur définition.

Un langage du premier ordre $\mathcal{L}$ est la donnée d'un ensemble dénombrable $\mathfrak{R}$ de symboles de prédicats. À chaque symbole de prédicat P, on associe un nombre naturel (éventuellement 0) qui spécifié le nombre d'arguments du symbole de prédicat. Ce nombre naturel est appelée *arité*.
L'ensemble $\mathcal{A}$ des formules atomiques est le plus petit ensemble contenant chaque expression $P(x_1, \ldots, x_n)$ où P est un symbole de prédicat d'arité $n \geqslant 0$, et $x_1, \ldots, x_n$ sont n variables. Finalement, l'ensemble $\mathcal{F}$ des formules du premier ordre est spécifié par la grammaire suivante :

$$\mathcal{F} := \mathcal{A} \mid \neg\mathcal{F} \mid \mathcal{F} \wedge \mathcal{F} \mid \mathcal{F} \vee \mathcal{F} \mid \mathcal{F} \to \mathcal{F} \mid \forall x\mathcal{F} \mid \exists x\mathcal{F}$$

Les variables libres et liées sont définies comme d'habitude. On peut supposer qu'on utilise deux alphabets : l'un pour les variables libres, l'autre pour les variables liées.

Une *interprétation* pour un langage du premier ordre $\mathcal{L}$, est $\mathfrak{M} = (M, -^{\mathfrak{M}})$ où M est un ensemble *non-vide* et $-^{\mathfrak{M}}$ est une fonction associant à chaque symbole de prédicat P avec arité n, un sous-ensemble $P^{\mathfrak{m}}$ du produit cartésien M^n de M.

Une *assignation* e est une fonction qui associe à certaines variables libre un élément de M : on note cela $e = [v_1 := a_1; \cdots; v_p := a_p]$, notation qui présuppose que les v_i soient des variables libres distinctes, et que les a_i soient des éléments de M, pas forcément distincts. On pourra écrire $e(v_i)$ pour a_i, et on notera $e \cup [y := b] = [v_1 := a_1; \cdots; v_p := a_p] \cup [y := b] = [v_1 := a_1; \cdots; v_p := a_p; y := b]$ lorsque y est une variable libre qui différente de chaque v_i.

On définit l'interprétation d'une formule H suivant l'assignation e définie pour chaque variable libre de H ainsi :

Finalement, si $\mathfrak{M} = (M, -^{\mathfrak{M}})$ est une interprétation pour un langage $\mathcal{L}$, F une formule de $\mathcal{L}$, et e une assignation des variables libres de F, on définit la notion "la formule F est vraie dans $\mathfrak{M}$ pour l'assignation e" que l'on écrit $\mathfrak{M}, e \models F$, par induction sur la structure de F :

- $\mathfrak{M}, e \models P(x_1, \ldots, x_n)$ ssi $(e(x_1), \ldots, e(x_n)) \in P^{\mathfrak{m}}$;
- $\mathfrak{M}, e \models \neg A$ ssi $\mathfrak{M}, e \not\models A$;
- $\mathfrak{M}, e \models A \wedge B$ ssi $\mathfrak{M}, e \models A$ et $\mathfrak{M}, e \models B$;
- $\mathfrak{M}, e \models A \vee B$ ssi $\mathfrak{M}, e \models A$ ou $\mathfrak{M}, e \models B$;
- $\mathfrak{M}, e \models A \rightarrow B$ ssi $\mathfrak{M}, e \models A$ implique $\mathfrak{M}, e \models B$;
- $\mathfrak{M}, e \models \forall x A$ ssi pour tout élément a de M, $\mathfrak{M}, e \cup [x := a] \models A$
- $\mathfrak{M}, e \models \exists x A$ ssi il existe un élément a de M tel que $\mathfrak{M}, e \cup [x := a] \models A$.

8.2 Traduction des logiques modales dans la logique du premier ordre

Soit $\mathcal{P} = \{p_1, \ldots, p_n, \ldots\}$ un ensemble de lettre propositionnelle d'un langage modale. Pour chacune de ces lettres propositionnelles, on se donne un symbole de prédicat unaire $P_1, \ldots P_n, \ldots$ dans un langage du premier ordre. Soit R un symbole de prédicat binaire. Soit $\mathcal{F}^{\mathfrak{m}}$ l'ensemble des formules modales engendrées par $\mathcal{P}$. Soit $\mathcal{F}$ l'ensemble des formules du premier ordre engendrées par $\{P_1, \ldots P_n \ldots\} \cup \{R\}$. On définit une fonction $\mathcal{F}^{\mathfrak{m}} \times \mathcal{V} \rightarrow \mathcal{F}$, associant à chaque formule modale F et variable x une formule du premier ordre $\boxed{F}_{@x}$, la traduction de F en x — x est une variable libre à laquelle sera assignée un monde possible. On appelle cette fonction *traduction standard*, et on la définie par induction sur la structure de la formule modale F

$$
\begin{aligned}
\boxed{p_i}_{@x} &= P_i(x) \\
\boxed{\neg A}_{@x} &= \neg \boxed{A}_{@x} \\
\boxed{A \wedge B}_{@x} &= \boxed{A}_{@x} \wedge \boxed{B}_{@x} \\
\boxed{A \vee B}_{@x} &= \boxed{A}_{@x} \vee \boxed{B}_{@x} \\
\boxed{A \rightarrow B}_{@x} &= \boxed{A}_{@x} \rightarrow \boxed{B}_{@x} \\
\boxed{\Box A}_{@x} &= \forall y \left(R(x,y) \rightarrow \boxed{A}_{@y} \right) \\
\boxed{\Diamond A}_{@x} &= \exists y \left(R(x,y) \wedge \boxed{A}_{@y} \right)
\end{aligned}
$$

On remarque aisément, par induction sur la formule que $\boxed{G}_{@x}$ n'a qu'une variable libre, à savoir x. En effet, lors de la traduction $\boxed{\Diamond A}_{@x}$ (ou $\boxed{\Box A}_{@x}$) une deuxième

variable apparaît, dans l'expression $\boxed{A}_{@_y}$, mais elle est liée.

La variable y est fraîche. Nous remarquons tout de suite que la traduction des formules modales suit la définition de satisfaction de formule modale dans un modèle de Kripke.

Étant donné un modèle de Kripke $\mathfrak{M} =< \mathcal{M}, \mathsf{R}, \Vdash_0 >$, on peut définir une interprétation du premier ordre $\mathfrak{M}' = (M', -^{\mathfrak{M}'})$ pour le langage du premier ordre mentionné ci-haut en posant :

— La base de modèle du premier ordre est l'ensemble des mondes du modèle de Kripke, c'est-à-dire $M' = \mathcal{M}$.
— pour tout P_i l'interprétation de P_i dans $\mathfrak{M}'$ est donné par l'ensemble des mondes qui sont en relation de forcing avec p_i, c'est-à-dire $P_i^{\mathfrak{M}'} = \{w \in M' \mid w \Vdash_0 p_i\}$
— L'interprétation du symbole binaire de prédicat R est donnée par l'ensemble de couples de mondes qui sont en relation par R, c'est-à-dire $R^{\mathfrak{M}'} = \{(w, w') \mid w\mathsf{R}w' \textit{ dans } \mathcal{M}\}$

On peut prouver la proposition suivante par induction sur la formule :

Proposition 53. *Soit* $\mathfrak{M} =< \mathcal{M}, \mathsf{R}, \Vdash_0 >$ *un modèle de Kripke et soit* $\mathfrak{M}'$ *le modèle du premier ordre correspondant. Soit A une formule modale et soit w un monde dans* $\mathcal{M}$*.*

$$w \Vdash A \quad ssi \quad \mathfrak{M}, e[x := w] \models \boxed{A}_{@x}$$

Chapitre 9

Logiques de Description

9.1 Une logique sans variables

Afin de décrire des données et de raisonner sur ces données les chercheurs ont fait apparaître des logiques sans variables inspirées des graphes conceptuels et de prolog/datalog.

Ces logiques sont appelées logiques de description. Une bonne partie des constructions mises en oeuvre dans ces formules leur permettent d'être vues à la fois comme des logiques multi modales propositionnelles et comme des fragments de la logique du premier ordre.

Ce chapitre s'appuie surtout sur [1], paru dans un handbook de logique modale, et qui met donc l'accent sur la vision multimodale des des logiques de description. Nous nous sommes aussi appuyé sur un autre article de synthèse, [9], plus orienté vers la logique du premier ordre et les applications des logique de descriptions.

La vision multimodale des logiques de description montre que les logiques de descriptions simples sont décidables, en appliquant simplement les résultats du chapitre 4 "décidabilité".

9.2 ALC : Attributive Language with Complements

ALC est un langage de description de concept. Un concept peut se voir comme une formule du premier ordre à une variable libre. La syntaxe de ALC permet de former des concept à partir de concept de base, et de prédicats binaires appelés rôles mais avec seulement certaines constructions de la logique du premier ordre, ce qui permet de ne PAS utiliser de variables.

On se donne un ensemble de concepts de base N (concept names), un ensemble de rôles R, et l'ensemble des concepts C définissables en ALC est défini comme suit :

$$C ::= N \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C$$

Du point de vue de la logique du premier ordre, les concepts de base de N sont des prédicats unaires (du langage) et les rôles R des prédicats binaires.

Si $A \in N$ est un concept de base, nous noterons aussi $A(\_)$ le prédicat unaire qui lui correspond. Il ya souvent un prédicat unaire $\bot$, et un prédicat $\top$.

Si $r \in R$ est un rôle, nous noterons aussi $r(\_,\_)$ le prédicat binaire qui lui correspond.

Nous pouvons traduire inductivement tout concept de ALC en une formule à une variable libre du langage du premier ordre ayant pour prédicats unaires les concept de base et pour prédicats binaires les rôles.

L'expression $\mathsf{fol}_x(C)$ où C est un concept de ALC, défini suivant la syntaxe ci-dessus désigne la traduction du concept en une formule du premier ordre à une variable libre x. Cette traduction est définie inductivement.

- $\mathsf{fol}_x(\bot) = (A(x)\&\neg A(x))$ (A concept de base arbitraire, l'important est que $\bot(x)$ soit faux pour chaque x dans toute interprétation)
- $\mathsf{fol}_x(\top) = (A(x) \rightarrow A(x))$ (A concept de base arbitraire, l'important est que $\bot(x)$ soit vrai pour chaque x dans toute interprétation)
- $\mathsf{fol}_x(A) = A(x)$
- $\mathsf{fol}_x(\neg C) = \neg(\mathsf{fol}_x(C))$
- $\mathsf{fol}_x(C \sqcap D) = \mathsf{fol}_x(C)\&\mathsf{fol}_x(D)$
- $\mathsf{fol}_x(C \sqcup D) = \mathsf{fol}_x(C) \vee \mathsf{fol}_x(D)$
- $\mathsf{fol}_x(\forall r.C) = \forall y.r(x,y) \rightarrow \mathsf{fol}_x(C)$
- $\mathsf{fol}_x(\exists r.C) = \exists y.r(x,y)\ \&\ \mathsf{fol}_x(C)$

Étant donné un domaine, une interprétation du langage du premier ordre associé les concepts complexes sont naturellement interprétés pas une partie du domaine, ceux qui satisfont la formule (ou qui font partie du concept). Cette interprétation est appelé la sémantique du concept.

En utilisant la correspondance déjà vue entre une logique modale la logique du premier ordre associée on peut traduire un langage ALC en une formule multimodale.

Pour chaque rôle r de R on se donne une paire de modalités : $\Box_r$ noté $[r]$ et $\Diamond_r$ noté $\langle r \rangle$. Pour chaque concept de base A, on se donne une proposition, notée A.

- $\mathsf{mod}(\bot) = \bot$
- $\mathsf{mod}(\top) = \top$
- $\mathsf{mod}(A) =$
- $\mathsf{mod}(\neg C) = \neg(\mathsf{mod}(C))$
- $\mathsf{mod}(C \sqcap D) = \mathsf{mod}(C)\&\mathsf{mod}(D)$
- $\mathsf{mod}(C \sqcup D) = \mathsf{mod}(C) \vee \mathsf{mod}(D)$
- $\mathsf{mod}(\forall r.C) = [r]\mathsf{mod}(C)$
- $\mathsf{mod}(\exists r.C) = \langle r \rangle \mathsf{mod}(C)$

Pour ce qui est des interprétations, il faut penser les mondes comme des individus, et l'interprétation d'un concept est l'ensemble des mondes où il est vrai, comme dans la traduction de la logique modale en logique du premier ordre.

Étant donnée une interprétation du premier ordre d'un langage ALC, il lui correspond un modèle de Kripke de la formule modale. La formule du premier ordre $\mathsf{fol}_x(C)[x := i]$ est vraie d'un individu i du domaine, si et seulement si le monde i force la formule modale correspondante, $i \Vdash \mathsf{mod}(C)$.

Par exemple, étant donné les concepts de base Femme, Voyou et le rôle parent, on peut définir un concept complexe "femme dont un enfant est un voyou qui n'aime aucun voyou" :

$$Femme \sqcap \exists parent.Voyou \& \forall aime.(\neg Voyou)$$

Ce qui s'écrit en logique du premier ordre ainsi :

$$Femme(x) \& \exists y(parent(x,y) \& Voyou(y)) \& \forall z(aime(x,z) \rightarrow \neg Voyou(z)$$

ou encore, en logique multimodale :

$$Femme \& \langle parent \rangle Voyou \& [aime] \neg Voyou$$

On remarquera que la présence des concept $\bot$ et $\top$ permet de définir les individus qui ne sont dans comme premier argument du rôle r avec aucun individu :

$$\forall r.\bot$$

c'est-à-dire

$$\forall y(R(x,y) \rightarrow \bot(y)) \equiv \forall y \neg R(x,y)$$

ou qui sont dans la relation r avec au moins un individu :

$$\exists r.\top$$

c'est-à-dire

$$\exists y(R(x,y) \& \top(y)) \equiv \exists y R(x,y)$$

9.2.1 Concepts individuels / nominaux

Il peut y avoir des concepts de bases ne sont valables que d'un seul individu. Leur interprétation du point de vue de la logique du premier ordre est nécessairement UN individu du domaine (comme s'il s'agissant d'une constante du langage), et du point de vue de la logique modale, l'interprétation d'un concept individuel consiste en UN monde possible.

Certaines définitions exigent que les interprétations des concepts individuels, aussi appelés concepts nominaux, soient différentes lorsque le nom n'est pas le même. C'est une différence par rapport à la logique classique, ou deux constantes peuvent fort bien être interprétées par le même individu du domaine d'interprétation. Si on procède ainsi on ne peut plus avoir d'axiomes $a \approx b$ ou $\neg(a \approx b)$ dans les ABix défnies ci-après.

9.3 Tbox

Une TBox est un semble d'axiomes d'inclusion de concepts : $C \subset D$, avec C, D deux concepts.

La formation des concepts peut utiliser les trois opérations suivantes, qui ne s'expriment pas aux moyens des opération vues dans la définition de ALC :

— $\exists r.Self$ qui correspond à $r(x,x)$ en logique du premier ordre (ou à $x \Vdash [r]H \to H$ pour toute formule H en logique modale)
— $\geqslant n\ r.C$ qui correspond à $\exists y_1 \cdots \exists y_n \bigwedge_{1\leqslant i,j\leqslant n, i\neq j} y_i \neq y_j \& \bigwedge_{1\leqslant i\leqslant n} r(x,y_i)$ — ce qui n'a pas d'équivalent direct en logique modale, c'est une condition sur la relation d'accessibilité r
— $\leqslant n\ r.C$ qui correspond à $\exists y_1 \cdots \exists y_n \exists y_{n+1} \bigwedge_{1\leqslant i\leqslant n+1} r(x,y_i) \multimap \bigvee_{1\leqslant i,j\leqslant n+1, i\neq j} y_i = y_j$ — ce qui n'a pas d'équivalent direct en logique modale, c'est une condition sur la relation d'accessibilité r

Une TBox est un ensemble d'inclusions de concepts appelés General Concept Inclusion (GCI),

$$C \sqsubseteq D$$

ce qui correspond à la formule du premier ordre $\forall x \mathsf{fol}_x(C) \to \mathsf{fol}_x(D)$ ou encore à $\mathsf{mod}(C) \to \mathsf{mod}(D)$. Un GCI est vrai dans un modèle du premier ordre lorsque $\forall x(\mathsf{fol}_x(C) \to \mathsf{fol}_x(D))$ est vraie dans ce modèle c'est-à-dire lorsque $\{x \in |M| \mid C[x]\} \subset \{x \in |M| \mid D[x]\}$; d'un point de vue modal, un GCI $C \sqsubseteq D$ est vrai dans un modèle de Kripke K lorsque pour tout monde possible i, on a $i \Vdash \mathsf{mod}(C) \to \mathsf{mod}(D)$.

9.3.1 Recursive TBox

9.4 Abox

Une ABox est un certains nombre d'assertion portant sur les concepts individuels notés $a, b, ...$, de la forme :

— $C[a]$ où $C[]$ est un concept — l'interprétation de a est dans celle de C pour la logique du premier ordre, ou la formule C est vraie dans le monde possible a.
— $r(a,b)$ où r est un rôle — le couple des interprétations de a et de b est dans l'interprétation de r pour la logique du premier ordre, ou il y a une relation d'accessibilité r du monde possible a vers le monde possible b, pour la logique multimodale.
— $\neg r(a,b)$ où r est un rôle — le couple des interprétations de a et de b n'est pas dans l'interprétation de r pour la logique du premier ordre, ou il n'y a pas de relation d'accessibilité r du monde possible a vers le monde possible b, pour la logique multimodale.
— $a \approx b$ (l'interprétation de a et de b sont un même individu du domaine pour la logique du premier ordre, ou les mondes possibles a et b sont le même pour la logique modale)
— $\neg(a \approx b)$ (l'interprétation de a et de b sont des individus différents du domaine pour la logique du premier ordre ; ou les mondes possibles a et b sont deux mondes différents pour la logique modale)

9.5 Rbox

Les RBox permettent d'exprimer des contraintes sur les rôles. On suppose qu'on a un rôle universel u et étant donné un rôle r on peut utiliser le rôle inverse qui sera noté, r^{-}.

Un axiome d'inclusion de rôle (RIA) est de la forme

$$r_1 \circ r_2 \circ \cdots \circ r_n \sqsubseteq r$$

dans lequel $\circ$ désigne la composition des rôles qui sont des relations binaires : $(x,z) \in r \circ s$ si et seulement $\exists y((x,y) \in s \& (y,z) \in r$.

Les inclusions de rôles simples sont de la forme $r \sqsubset s$, avec r, s des rôles.

L'ensemble des RIA est appelée une hiérarchie de rôles.

Il ya des conditions de non circularité des $\sqsubseteq$ afin que la logique qui en résulte reste décidable.

9.6 Résultats

9.7 Exercices

Exercice 54. *1. Montrez que la logique du premier ordre peut exprimer qu'un prédicat binaire complexe ou faisant partie du langage (un rôle) est transitif.*

2. *Montrez qu'il n'y a pas de formule du premier ordre, à deux variables libres x et y exprimant que x,y est dans la clôture transitive d'un prédicat binaire complexe ou faisant partie du langage (un rôle).*
3. *Peut-on exprimer en logique (multi)modale qu'une relation d'accessibilité est transitive et si oui, comment faire ?*

Chapitre 10

Sémantique du langage naturel : logique modale du premier ordre

Bibliographie

[1] F. Baader & C. Lutz – « Description logic », in *Handbook of Modal Logic* (P. Blackburn, J. F. A. K. van Benthem & F. Wolter, éds.), Studies in logic and practical reasoning, vol. 3, North-Holland, 2007, p. 757–819.

[2] J. van Benthem – *Modal logic for open minds*, CSLI Publications, 2010.

[3] P. Blackburn, M. de Rijke & Y. Venema – *Modal logic*, Cambridge Tracts in Theoretical Computer Science, vol. 53, Cambridge University Press, 2001.

[4] S. Demri, V. Goranko & M. Lange – *Temporal logics in computer science : Finite-state systems*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2016.

[5] H. van Ditmarsch & B. Kooi – *One hundred prisoners and a light bulb*, Springer, 2015.

[6] J. Duparc – *La logique pas-à-pas*, Presses Polytechniques et Universitaires Romandes, 2015.

[7] R. Fagin, J. Y. Halpern, Y. Moses & M. Y. Vardi – *Reasoning about knowledge*, MIT Press, 1995.

[8] M. Fitting & R. Mendelsohn – *First-order modal logic*, First-order Modal Logic, Springer Netherlands, 1998.

[9] S. Rudolph – « Foundations of description logics », in *Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures* (A. Polleres, C. d'Amato, M. Arenas, S. Handschuh, P. Kroner, S. Ossowski & P. F. Patel-Schneider, éds.), Lecture Notes in Computer Science, vol. 6848, Springer, 2011, p. 76–136.

Annexe A

Rappels de logique classique propositionnelle

A.1 Langage

Le langage propositionnel est défini comme suit, où *Prop* désigne un ensemble de variable propositionnelles (on dit aussi littéraux, lettres,...)

$$L ::= Prop | \neg L | L \& L | L \vee L | L \multimap L$$

A.2 Interprétation, valuation, théorie

Une valuation ou interprétation est une application $v : Prop \mapsto \mathcal{B} = \{0, 1\}$ qui envoie les variables propositionnelles sur la valeur 0 ou 1.

Ainsi toute formule reçoit une valeur 1 ou 0 — mais jamais les deux à la fois ! (Induction facile).

Définition 23. *Une théorie est un ensemble* pas forcément fini *de formules. Une théorie finie* $\mathsf{Th} = \{G_1, \ldots, G_n\}$ *est équivalente à la formule* $G_1 \wedge \cdots \wedge G_n$*. Une théorie infinie* $\{G_i | i \in I\}$ *avec I infini n'est équivalente à* aucune *formule.*

Une théorie T est dite contradictoire *ou* insatisfiable *s'il n'existe pas de valuation qui rende chaque formule de T vraie (qui satisfasse toute formule de T) Dans le cas contraire, T est dite* consistante .

Une formule C est conséquence sémantique *d'une théorie T si et seulement toute valuation satisfaisant toutes les formules de T satisfait C.*

Si C est conséquence sémantique de C' *et réciproquement, on dit que C et* C' *sont* sémantiquement équivalentes.

A.3 Preuves à la Hilbert et validité

Une démonstration de la formule ϕ dans une théorie T est une suite **finie** de n formules $\phi_1, \ldots, \phi_{n-1}, \phi_n$ dont la dernière est $\phi_n = \phi$ et dont chaque ligne ϕ_i, $i \leqslant n$ est soit

- une formule de T : $\phi_i \in T$
- un instance d'un axiome propositionnel
- une formule $\phi_i = C$ obtenue par modus ponens à partir de 2 lignes déjà obtenues $\phi_k = A \Rightarrow C$, $\phi_j = A$ avec $j < i$ et $k < i$

Axiomes du calcul propositionnel :

- $S = (p \multimap q \multimap r) \multimap (p \multimap q) \multimap (p \multimap r)$
- $K = p \multimap q \multimap r$
- $I = p \multimap p$ (se déduit de K et S)
- $(\neg p \multimap \neg q) \multimap ((\neg p \multimap q) \multimap p)$
- $p \multimap q \multimap (p \wedge q)$
- $(p \wedge q) \multimap p$
- $(p \wedge q) \multimap q$
- $p \multimap (p \vee q)$
- $q \multimap (p \vee q)$
- $(p \multimap r) \multimap (q \multimap r) \multimap (p \vee q) \multimap r$

Une formule est conséquence de la théorie T lorsque $T \vdash C$ c'est-à-dire il existe une démonstration formelle de conclusion C à partir de T.

Théorème 55 (Lemme de la déduction)**.** *(*) $T \cup \{H\} \vdash G$ si et seulement si $T \vdash H \multimap G$ (**)*

Démonstration. (**) entraîne (*) est évident par modus ponens.

(*) entraîne (**) étant donnée une preuve (suite de formules) $G1, \ldots, G_n = G$ utilisant H en plus de T transformons la, ligne par ligne, en une preuve $H \multimap G_1, \ldots, H \multimap G_n = G$ n'utilisant pas H en intercalant si besoin quelques lignes entre $H \multimap G_i$ et $H \multimap G_{i+1}$

- Si $G_i = H$ on a $H \multimap H$ est un axiome.
- Si G_i est une instance d'un axiome ou une formule de T on place l'axiome $G_i \multimap (H \multimap G_i)$ (K) entre $H \multimap G_{i-1}$ et par modus ponens on obtient $H \multimap G_i$
- Si G_i était obtenu par modus ponens à partir de
  - G_j avec $j < i$
  - $G_k = G_j \multimap G_i$ avec $k < i$

  Dans la nouvelle preuve"transformée" on a les lignes $H \multimap G_j$ et $H \multimap G_j \lim G_i$, avant la transformation de la ligne G_i on insère l'axiome $S : (H \multimap G_j \multimap G_i) \multimap (H \multimap G_j) \multimap (H \multimap G_i)$ et avec deux modus ponens, on obtient la ligne $H \multimap G_i$.

□

Théorème 56 (Validité cohérence soundness du système déductif de Hilbert)**.** *S'il existe une preuve dans un système à la Hilbert de* $\mathsf{Th} \vdash C$ *où* Th *est une théorie, alors pour toute valuation v qui rend vraie toutes les formules de* Th*, on a $v(C) = 1$.*

Démonstration. Il est aisé de voir que les axiomes sont vraies quelles que soient les valeurs des variables propositionnelles, et que le modus ponens préserve la vérité : si $v(A) = 1$ et si $v(A \multimap B) = 1$ alors $v(B) = 1$. DOnc si on suppose les formules de Th vraies, si $\mathsf{Th} \vdash C$, on a $v(C) = 1$. □

A.4 Calcul des séquents propositionnel LK_0 et validité

Les suites *finies* de formules (possiblement vides) sont notées par des majuscules grecques : $\Gamma \vdash \Delta$.

Un séquent noté $\Gamma \vdash \Delta$ est composé de deux suites de formules Γ et Δ séparées par le signe "$\vdash$". Les formules de Γ sont appelées les hyptohèses du séquents tandis que les formules de Γ sont appelés les conclusions du séquent.

Etant données une valuation v la valeur d'un séquent

$$H_1, \ldots, H_n \vdash C_1, \ldots, C_p$$

est

$$v([H_1 \wedge \cdots \wedge H_n] \multimap [C_1 \vee \cdots \vee C_p]) = \neg v(H_1) \vee \cdots \vee \neg v(H_n) \vee v(C_1) \vee \cdots \vee v(C_p)$$

Si $p = n = 0$ le séquent vaut 0.

L'axiome (on peut se limiter à A variable propositionnelle) :

$$\Gamma, A \vdash A, \Delta$$

Règles

$$\frac{\Gamma, A, B \vdash \Theta}{\Gamma, (A \wedge B) \vdash \Theta} \wedge_g \qquad \frac{\Theta \vdash A, \Delta \qquad \Theta \vdash B, \Delta}{\Theta \vdash (A \wedge B), \Delta} \wedge_d$$

$$\frac{\Gamma, A \vdash \Theta \qquad \Gamma, B \vdash \Theta}{\Gamma, (A \vee B), \vdash \Theta} \vee_g \qquad \frac{\Theta \vdash A, B, \Delta}{\Theta \vdash (A \vee B), \Delta} \vee_d$$

$$\frac{\Gamma \vdash A, \Theta \qquad \Gamma, B \vdash \Theta}{\Gamma, (A \multimap B) \vdash \Theta} \multimap_g \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash (A \multimap B), \Delta} \multimap_d$$

$$\frac{\Gamma \vdash A, \Theta}{\Gamma, (\neg A) \vdash \Theta} \neg_g \qquad \frac{\Gamma, A \vdash \Theta}{\Gamma \vdash (\neg A), \Theta} \neg_d$$

Théorème 57 (Validité cohérence soundness de LK_0)**.** *Le calcul LK_0 est valide (on dit aussi cohérent, en anglais "sound") : les séquents démontrés sont vrais pour toute valuation.*

Démonstration. Les axiomes sont vrais pour toute valuation (la formule en commun ne pouvant être à la fois vraie et fausse pour une valuation donnée) et les règles préservent la validité. □

La propriété suivante est essentielle pour la recherche de preuve dans Lk_0 :

Proposition 58 (Propriété de la sous formule)**.** *LK_0 (sans autre règle) a la propriété de la sous formule : toute formule d'un (des) séquent(s) prémisse(s) d'une règle est sous-formule d'une formule du séquent conclusion de cette règle.*

Ce n'est pas le cas pour la règle de coupure, pourtant utile puisque cette règle correspond à l'utilisation d'un lemme :

$$\frac{\Theta \vdash A, \Delta \qquad \Theta, A \vdash \Delta}{\Theta \vdash \Delta}\ cut$$

Proposition 59. *La règle de coupure préserve aussi la validité.*

Théorème 60. *Toute démonstration de LK_0 enrichie par la coupure peut-être transformée en une preuve de LK_0, qui n'utilise pas la règle de coupure et qui satisfait donc la propriété de la sous formule.*

Démonstration. Le fait qu'un séquent démontrable avec coupure le soit sans coupure découle de la *preuve* du théorème de complétude du calcul propositionnel ci-après (théorème 3) en utilisant l'agorithme de recherche de preuve : la recherche de preuve n'utilise pas la règle de coupure. □

Exemple 61. *Un exemple de preuve :*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{q,p,p,s \vdash q,t,t}{q,p,(\neg q),p,s \vdash t,t}\neg_g}{q,p,(\neg q),(p \wedge s) \vdash t,t}\wedge g \qquad \dfrac{\dfrac{r,q,p,p,s \vdash q,t}{r,q,p,(\neg q),p,s \vdash t}\neg g}{r,q,p,(\neg q),(p \wedge s) \vdash t}\wedge g}{q,p,(\neg q),(t \multimap r),(p \wedge s) \vdash t}\multimap g \qquad p,(\neg q),(t \multimap r),(p \wedge s) \vdash p,t}{p,(\neg q),(p \multimap q),(t \multimap r),(p \wedge s) \vdash t}\wedge g}{(p \wedge (\neg q)),(p \multimap q),(t \multimap r),(p \wedge s) \vdash p,t}\wedge g$$

Proposition 62. *L'algorithme de recherche de preuve d'un séquent Gamma $\vdash \Delta$ qui consiste à choisir une formule de Γ ou de Δ et à remonter la règle du connecteur principal de cette formule, termine par des séquents atomiques, ne contenant que des variables propositionnelles, qui peuvent être des axiomes ou non.*

Démonstration. Évident, par induction sur le nombre de connecteurs présents dans un séquent : chaque séquent prémisse de la règle a au moins un connecteur de moins que le séquent conclusion de la règle. □

Exemple 63. *Un exemple de preuve obtenue par recherche de preuve :*

$$\dfrac{\dfrac{\dfrac{\mathbf{q}, p, p, s \vdash \mathbf{q}, t, t}{q, p, (\neg q), p, s \vdash t, t}\,\neg_g}{q, p, (\neg q), (p \wedge s) \vdash t, t}\,\wedge g \qquad \dfrac{\dfrac{r, \mathbf{q}, p, p, s \vdash \mathbf{q}, t}{r, q, p, (\neg q), p, s \vdash t}\,\neg_g}{r, q, p, (\neg q), (p \wedge s) \vdash t}\,\wedge g}{q, p, (\neg q), (t \multimap r), (p \wedge s) \vdash t}\,\multimap_g$$

Exemple 64. *Un exemple d'application d'application de l'algorithme de recherche de preuve ne conduisant pas à une preuve ($p \vdash r, s$ n'est pas un axiome).*

$$\dfrac{\dfrac{\dfrac{\dfrac{p \vdash r, s}{\vdash r, s, (\neg p)}\,\neg_d}{(\neg r) \vdash s, (\neg p)}\,\neg_g}{\vdash (\neg p), ((\neg r) \multimap s)}\,\multimap_d \qquad \dfrac{\dfrac{q \vdash r, s}{(\neg r), q \vdash s}\,\neg_g}{q \vdash ((\neg r) \multimap s)}\,\multimap_d}{((\neg p) \multimap q) \vdash ((\neg r) \multimap s)}\,\multimap_g}{\vdash ((\neg p) \multimap q) \multimap ((\neg r) \multimap s)}\,\multimap_d$$

A.5 Equivalence du calcul des séquents et du système à la Hilbert

Proposition 65. *Si une théorie* Th *démontre C dans le calcul à la Hilbert, alors il existe une partie finie* Th_f *de Th telle que le séquent* $\mathsf{Th}_f \vdash C$ *est démontrable dans le calcul des séquents.*

Démonstration. La preuve à la Hilbert étant une suite *finie* de formules, elle n'utilise qu'un nombre fini de formules de Th. Appelons les Th_f. On peut dériver les instances des axiomes de Hilbert et la règle de modus ponens dans le calcul des séquents. On peut alors montrer par induction sur le nombre d'étapes de la preuve à la Hilbert que $\mathsf{Th}_f \vdash C$. □

Proposition 66. *S'il existe une démonstration d'un séquent $\Gamma \vdash C_1, \ldots, C_n$ alors le système de Hilbert démontre $C_1 \vee \cdots \vee C_n$ dans la théorie Γ.*

Démonstration. On procède par induction sur les dérivations du calcul des séquents, qui sont transformées en déductions à la Hilbert. Ce n'est pas aussi facile que dans l'autre sens. □

Théorème 67 (Equivalence H LK)**.** *Soit* Th *une théorie. Le système déductif de Hilbert démontre C si et seulement s'il existe une partie finie* Th_f *de* Th *telle que le calcul des séquents démontre* $\mathsf{Th}_f \vdash C$.

En d'autres termes, $\mathsf{Th} \vdash C$ *à la Hilbert si et seulement si* $\mathsf{Th} \vdash C$ *dans le calcul des séquents — $\mathsf{Th} \vdash C$ dans le calcul des séquents est défini par il existe une partie finie* Th_f *de* Th *telle que* $\mathsf{Th}_f \vdash C$.

Démonstration. C'est la conséquence directe des deux propositions précédentes. □

A.6 Complétude du calcul des séquents LK_0

La complétude du calcul des séquents LK_0 résulte d'une propriété particulière de LK_0 :

Proposition 68 (réversibilité de LK_0)**.** *Chaque de LK_0 est réversible en ce sens que si le séquents conclusion est vrai pour une valuation v, alors le(s) séquent(s) prémisses de la règle sont vrais pour v.*

En d'autres termes, si l'un au moins des séquents prémisses d'une règle de LK_0 est faux pour une valuation v, alors le séquent conclusion est aussi faux pour cette valuation v.

Démonstration. Il suffit de vérifier la réversibilité de chaque règle. □

Théorème 69. *Si un séquent est vrai pour toute valuation, alors il est démontrable.*

Démonstration. Montrons plutôt la contraposée : un séquent $H_1,\dots,H_n \vdash C_1,\dots,C_p$ n'est pas démontrable alors il existe une valuation telle que ce séquent soit faux.

Appliquons l'algorithme de recherche de preuve ci-dessus (cf. proposition 62). Celui-ci se termine par des séquents atomiques, dont les formules sont toutes, à droite comme à gauche du signe "$\vdash$" des variables propositionnelles. Comme ce séquent n'est pas démontrable, l'un au moins de ces séquents atomiques disons $h_1,\dots,h_k \vdash c_1,\dots,c_l$ n'est pas un axiome, c'est-à-dire s'il ne contient pas la même variable propositionnelle des deux côtés ($h_i \neq c_j$ pour tous i et j), on peut définir une valuation ainsi : $w(h_i) = 1$ et $w(c_j) = 0$ en fixant arbitrairement $v(p)$ pour les variables propositionnelles p qui ne sont ni des h_i ni des c_j — il n'y aurait pas de telle valuation si ce séquent comportait une variable commune à sa partie gauche et à sa partie droite, c'est-à-dire si $h_{i_0} = c_{j_0} = q$, car il faudrait que $w(q) = 1$ et $w(q) = 0$! En vertu de la réversibilité de LK_0 (cf. proposition 68 on obtient que $w(H_1,\dots,H_n \vdash C_1,\dots,C_p) = 0$. □

A.7 Complétude du système déductif de Hilbert

Remarque 70. *Etant donné un langage propositionnel, c'est-à-dire un ensemble fini ou énumérable de variables propositionnelles $\mathcal{P}$, les formules propositionnelle sur ce langage sont énumérables $G_1, G_2, \dots$*

Définition 24. *Soit* Th *une théorie (un ensemble par forcément fini de formules), on dit que* Th *démontre la formule C notation* $\mathsf{Th} \vdash C$ *s'il existe une partie* finie $\{T_1,\dots,T_n\}$ *de* Th *telle que* $T1,\dots,T_n \vdash C$ *(dans le calcul des séquents, ou dans le système de Hilbert). On dit que* Th *est incohérente lorsque* $T \vdash \bot$*, ou, ce qui revient au même, lorsqu'il existe une formule G telle que* $\mathsf{Th} \vdash G$ *et* $\mathsf{Th} \vdash \neg G$*. On dit que* Th *est cohérente lorsque* Th *n'est pas cohérente.*

Définition 25. *Une théorie cohérente est dite complète si et seulement si pour toute formule H soit $H \in \mathsf{Th}^+$ soit $\neg H \in \mathsf{Th}^+$ (et pas les deux, évidemment, car elle est cohérente).*

Proposition 71. *Si une théorie cohérente* Th *ne démontre pas* G*, alors* $\mathsf{Th} \cup \{\neg G\}$ *est cohérente.*

Démonstration. Montrons la contraposée. Si $\mathsf{Th} \cup \{\neg G\}$ est incohérente, $\mathsf{Th} \cup \{\neg G\} \vdash \bot$ et par lemme de la déduction $\mathsf{Th} \vdash \neg G \multimap \bot$, et comme $\neg G \multimap \bot \vdash G$ (dans le système de Hilbert tout comme dans le calcul des séquents), on a $\mathsf{Th} \vdash G$. □

Proposition 72. *Si* Th *est cohérente et complète et si* $\mathsf{Th} \vdash A$ *alors* $A \in \mathsf{Th}$*.*

Démonstration. Si Th est complète alors pour toute formule G on a $G \in \mathsf{Th}$ ou $\neg G \in \mathsf{Th}$. Si $\mathsf{Th} \vdash A$, alors $\neg A \notin \mathsf{Th}$ — en effet, si $\neg A \in \mathsf{Th}$ nous aurions $\mathsf{Th} \vdash \neg A$ et avec $\mathsf{Th} \vdash A$ bous aurions $\mathsf{Th} \vdash \bot$, ce qui est impossible car Th est cohérente. □

Proposition 73. *Une théorie cohérente est complète si et seulement si elle est maximale pour l'inclusion.*

Démonstration. Supposons qu'une théorie Th soit maximale pour l'inclusion, montrons qu'elle est complète.

— Si $\mathsf{Th} \nvdash G$, alors on a vu que $\mathsf{Th} \cup \{\neg G\}$ est cohérente (proposition 71). Comme Th maximale, $\neg G \in \mathsf{Th}$.

— Sinon, c'est-à-dire si $\mathsf{Th} \vdash G$, $\mathsf{Th} \vdash G$ cohérente, et comme Th maximale, $G \in \mathsf{Th}$.

Maintenant supposons qu'une théorie Th soit cohérente et complète et montrons qu'elle est maximale pour l'inclusion. Considérons une formule H qui ne soit pas dans Th. Comme Th est complète $\neg H$ est dans Th et on ne peut pas ajouter H à Th en préservant la cohérence. □

Proposition 74. *Toute théorie cohérente* Th *admet une extension cohérente et complète.*

Démonstration. Énumérons toutes les formules du langage propositionnel considéré : $(G_i)_{i\in\mathbb{N}}$ — c'est possible car nous ne considérons que des langages dont les variables propositionnelles sont énumérables. Posons $\mathsf{Th}_0 = \mathsf{Th}$.

Pour chaque i, définissons Th_{i+1} ainsi. Si $\{G_i\} \cup \mathsf{Th}_{i+1}$ cohérente, $\mathsf{Th}_{i+1} = \{G_i\} \cup \mathsf{Th}_{i+1}$ sinon $\mathsf{Th}_{i+1} = \mathsf{Th}_i$.

Appelons Th^+ l'union $\bigcup_{i\in\mathbb{N}} \mathsf{Th}_i$.

Montrons que Th^+ est cohérente. Si elle ne l'était pas nous aurions une preuve de $\mathsf{Th}^+ \vdash \bot$. Comme une preuve est finie, cette preuve n'utilise qu'un nombre fini de formules de Th et des G_i ajoutées à Th. Ces formules en nombre fini appartiennent toutes à un même Th_K, et il est aisé, par récurrence sur i de voir que tous les Th_i sont cohérentes.

Montrons maintenant que Th^+ est maximal. Sinon nous pourrions rajouter l'une des formules, disons G_l de l'énumération. Si elle n'a pas été rajoutée à Th_l pour former Th_{l+1} c 'est que $\mathsf{Th}_l \cup \{G_l\}$ n'est pas cohérente. A fortiori, l'ajout de la formule G_l à $\mathsf{Th}^+ \supset \mathsf{Th}_l$ serait incohérente. □

Proposition 75. *Pour théorie cohérente et complète* Th *il existe une et un seule valuation qui rendre vraie toute formule de de* Th *et fausse toute formule qui ne soit pas dans* Th*.*

Démonstration. La valuation v_{Th} est définie ainsi : pour chaque variable propositionnelle p qui est dans Th on pose $v_{\mathsf{Th}}(p) = 1$ et pour chaque variable propositionnelle q qui n'est pas dans Th on pose $v_{\mathsf{Th}}(q) = 0$. Il est aisé de voir par induction sur la taille de la formules qu'une formule $H \in \mathsf{Th}$ si et seulement si $v_{\mathsf{Th}}(H) = 1$ (on remarque que $v_{\mathsf{Th}}(H) \neq 1$ équivaut à $v_{\mathsf{Th}}(H) = 0$.

L'équivalence est vraie lorsque H est une variable propositiopnnelle et ce par définition de v_{Th}.

Supposons que l'équivalence soit vraie pour toute les formules de taille inférieure ou égale à n. Soit H une formule de hauteur $n+1$, montrons que l'équivalence est vrai pour H. Il faut regarder ce qui se passe pour chaque connecteur. Traitons ici le cas $H = A \multimap B$, les autres étant plus simples.

Si $H = A \multimap B$, alors A et B sont de taille inférieure ou égale à n.

— Montrons pour commencer que $v_{\mathsf{Th}}(A \multimap B) = 1$, alors $A \multimap B \in \mathsf{Th}$. Si $v_{\mathsf{Th}}(A \multimap B) = 1$ ce la signifie que nous sommes dans au moins l'un des deux cas suivants :
  — $v_{\mathsf{Th}}(A) = 0$ Comme A est de taille inférieure à n, on peut lui appliquer l'hypothèse d'induction, donc $A \notin \mathsf{Th}$. Comme Th est complète $\neg A \in \mathsf{Th}$, et comme $\neg A \vdash A \multimap B$ on a $A \multimap B \in \mathsf{Th}$ (cf. proposition 72).
  — $v_{\mathsf{Th}}(B) = 1$ et donc comme B est plus petite que $A \multimap B$, $B \in \mathsf{Th}$, et comme $B \vdash A \multimap B$, $A \multimap B \in \mathsf{Th}$ (cf. proposition 72).
— Montrons maintenant que si $wth(A \multimap B) = 0$, alors $A \multimap B \notin \mathsf{Th}$. Si $wth(A \multimap B) = 0$, cela signifie que $v_{\mathsf{Th}}(A) = 1$ et que $v_{\mathsf{Th}}(B) = 0$. Comme A et B sont plus petites que $A \multimap B$, on a par hypothèse d'induction que $A \in \mathsf{Th}$ et $B \notin \mathsf{Th}$, et donc $\neg B \in \mathsf{Th}$. Donc $A \in \mathsf{Th}$ et $\neg B \in \mathsf{Th}$, et par conséquent $A \wedge \neg B \in \mathsf{Th}$, et $A \multimap B\neg \in \mathsf{Th}$, car $A \multimap B, A\neg B \vdash \bot$.

L'unicité est évidente, il suffit de remarque que chaque variable propositionnelle est dans Th ou n'y est pas. □

Théorème 76. *Le système déductif de Hilbert est complet pour la logique propositionnel : étant données une théorie cohérente* Th*, et une formule G si toutes les valuations qui rendent toutes les formules de* Th *vraies, rendent aussi G vraie, alors* $\mathsf{Th} \vdash G$ *dans le système déductif à la Hilbert.*

Démonstration. Supposons que $\mathsf{Th}\neg \vdash G$. On a alors $\mathsf{Th} \cup \neg G$ est cohérente (proposition 71), et admet une extension complète et cohérente (proposition 74), notons la Th^{ng+}. Considérons la valuation w définie comme ci dessus par $w(q) = 1$ si $q \in \mathsf{Th}^{ng+}$, et $w(q) = 0$ si $q \notin \mathsf{Th}^{ng+}$. Pour toute formule de $H \in \mathsf{Th}^{ng+}$, on a $w(H) = 1$. On a donc en particulier $w(H) = 1$ pour toute formule H de $\mathsf{Th} \subset \mathsf{Th}^{ng+}$ et $w(\neg G) = 1$ car $\neg G \in \mathsf{Th}^{ng+}$. Donc $w(G) = 1$. Nosu avons donc une valuation w qui rend toutes les formiules de *th* vraies, et G fausse, ce qui contredit l'hypothèse selon laquelle dans toute valuation où toutes les formules de Th sont vraies G est vraies. Par conséquent il est faux que $\mathsf{Th} \nvdash G$ et donc $\mathsf{Th} \vdash G$. □

A.8 Conséquences de la validité et de la complétude de LK_0 et H

A.8.1 Décidabilité

Théorème 77. *La validité d'une formule du calcul propositionnel est un problème décidable.*

Démonstration. Une formule est valide si et seulement si elle est vraie pour toute valuation. Les tables de vérité permette de vérifier si la formule est valide : s'il n'y a que des que des 1 sur chaque ligne elle est valide et s'il y a un 0 sur une ligne particulière, il y a une valuation pour laquelle elle est fausse. Si la formule utilise n variable propositionnelles, la table a 2^n sur chaque ligne et autant de colonnes que de sous formules.

D'après le théorème de complétude, cette question revient à montrer que le séquent composé de la formule à droite du signe "$\vdash$" est démontrable. L'algorithme de recherche de preuve termine, soit par des axiomes et donc par une preuve et la formule et valide, soit par au moins un séquent atomique qui n'est pas un axiome et alors la formule peut être rendu fausse par une valuation qui rend fausse cet axiome. □

A.8.2 Compacité du calcul propositionnel

Théorème 78. *Soit* Th *une théorie telle que toute partie finie* Th_f *admette une valuation* v_{Th_f} *qui rende chaque formule de* Th_f *vraie. Alors il existe une valuation qui rend toutes les formules de* Th *simultanément vraies.*

Démonstration. En vertu du théorème de complétude (théorème 3), si Th n'admet pas de valuation qui rende toutes ses formules vraies, Th est incohérente. Si Th est incohérente il existe une preuve de $\mathsf{Th} \vdash \bot$. Cette preuve, étant fini comme l'est toute preuve, ne fait appel qu'à un nombre fini de formules de Th, disons Th_f. Donc $\mathsf{Th}_f \vdash \bot$, et cela contredit la validité du système déductif utilisé (calcul des séquents ou système de Hilbert). □

A.8.3 Equivalence du calcul des séquents et du système de Hilbert

On peut voir l'équivalence de ces deux systèmes déductifs "règle à règle", comme ci-dessus mais on peut aussi utiliser la validité et la complétude de chacun de ces deux systèmes déductifs :

— Soit une formule H démontrable à la Hilbert dans la théorie Th c'est-à-dire $\mathsf{Th} \vdash C$ à la Hilbert. Cette preuve utilise un nombre fini de formules Th_f de Th. Par validité du système de Hilbert (théorème 56) toute valuation rendant vrai Th_f rend C vraie. Par complétude du calcul des séquents LK_0 le séquent $\mathsf{Th}_f \vdash C$ est démontrable dans LK_0, et donc $\mathsf{Th} \vdash C$ dans LK_0.

— Si on a $\mathsf{Th} \vdash C$ dans le calcul des séquents, cela signifie qu'il existe une partie finie Th_f de Th telle que le séquent $\mathsf{Th}_f \vdash C$ soit démontrable. Comme le calcul des séquent est valide, toute valuation qui rend Th_f vraie rend C vraie. Le système de Hilbert étant complet, il existe une preuve dans le système de Hilbert de C dans $\mathsf{Th}_f \subset \mathsf{Th}$, et donc a fortiori une preuve de C dans Th.

A.8.4 Elimination des coupures

Comme dit ci-dessus un séquent qui est démontrable dans LK_0 enrichi pas la règle de coupure peut l'être dans LK_0 sans la règle de coupure. Une preuve consiste à transformer algorithmiquement la preuve avec coupure en une preuve sans coupure, c'est une induction assez complexe. Si on ne s'intéresse qu'à la possibilité, il suffit de remarquer que la règle de coupure étant valide, un séquent prouvable est vrai pour toute valuation. Par complétude de LK_0, ce séquent est donc démontrable dans LK_0 : l'agorithme de recherche de preuve n'utilise pas la règle de coupure, termine et donne une preuve, sinon il y aurait une valuation qui rendrait ce séquent faux.

A.9 Exercices

Exercice 79. *Soit $\mathcal{P} = \{p_1, \ldots, p_n, \ldots\}$ un ensemble de variables propositionnelles, soit T une théorie sur ce langage.*

1. *Montrer que si T ne démontre pas la formule G alors $T \cup \neg G$ est cohérente. Vous énoncerez clairement les théorèmes utilisés.*
2. *Montrer que si T ne démontre ni G ni $\neg G$ alors il existe des valuations pour lesquelles G est vraie et d'autres pour lesquelles G est fausse. Vous énoncerez clairement les théorèmes utilisés.*
3. *Connaissez-vous des exemples de théories infinies T telles que, pour certaines formules G, T ne démontre ni G ni $\neg G$? Si oui, connaissez-vous un exemple de telle formule G ?*

Exercice 80 (Calcul des séquents — validité/cohérence de LK_0 (soundness))**.**

1. *Montrez que les axiomes de LK_0 sont valide pour toute valuation.*
2. *Montrez que les règles de LK_0 sont valide (si le(s) séquent(s) prémisses sont vrais pour une valuation $v(\cdot)$ alors la conclusion est vraies pour cette même valuation. [On traitera quelques règles autres que celles vues en cours]*
3. *Montrez que le calcul des séquents est valide en ce sens que si LK_0 démontre $\Gamma \vdash \Delta$ alors $\Gamma \vdash \Delta$ est vrai pour toute valuation.*

Exercice 81 (Calcul des séquents — règles de LK_0 inversibles)**.**

1. *Montrez qu'il existe une valuation rendant faux un séquent atomique (ne contenant que les variables propositionnelles sans connecteur) dont les parties droite et gauche ne contiennent pas de variables propositionnelles communes (qui n'est pas un axiome).*
2. *Montrez que les règles de LK_0 sont réversibles (si un des séquent(s) prémisse(s) est faux pour une valuation $v(\cdot)$ alors la conclusion est fausse pour cette même valuation. [On traitera quelques règles autres que celles vues en cours]*
3. *Montrez que si l'agorithme de recherche des preuves à partir du séquents $\Gamma \vdash \Delta$ conduit à un séquent atomique qui n'est pas un axiome, alors il existe une valuation qui rend le séquent faux.*

Exercice 82 (Calcul des séquents — recherche de preuve)**.** *Montrer que l'algorithme de recherche de preuve termine toujours dans le calcul des séquents LK_0 (à la différence de LK). En déduire q'un séquent est soit démontrable et valide soit non démontrable et faux pour une certaine valuation. En déduire que la règle de coupure est redondante dans le calcul des séquents.*

Exercice 83. *Montrer que les axiomes de Hilbert sont démontrables dans le calcul des séquents. Montrer que la règle de modus ponens est simulable dans le calcul des séquents.*

Exercice 84.

1. *Montrez que $p \multimap p$ dans un système à la Hilbert.*
2. *Montrez le lemme de la déduction.*
3. *Montrer que l'axiome du calcul des séquents est démontrable dans un système à la Hilbert.*
4. *Montrer que les règles du calcul des séquents sont simulables dans un système à la Hilbert. [Ne faire qu'une ou deux règles.]*

Exercice 85. *Déduire des deux exercices précédents, que*

1. *le calcul des séquents et le système déductif à la Hilbert démontrent exactement les mêmes formules,*
2. *toute formule vraie pour toute valuation est démontrable dans le calcul des séquents sans coupure et dans les système à la Hilbert*
3. *le calcul des séquents avec coupure et sans coupure démontrent exactement les mêmes séquents.*

Annexe B

Rappels de logique classique du premier ordre

Annexe C

Rappels sur la logique intuitionniste