# Logiques modales

Jean Goubault-Larrecq

March 5, 1997

## 1 Introduction et motivation

Les logiques modales, dont les logiques temporelles sont des cas particuliers, ont été proposées au début du vingtième siècle par Clarence Lewis [Lew18]. Lewis n'était pas satisfait de l'implication matérielle de la logique classique, et a exploré une autre variété de logiques, où un nouvel opérateur $\Box$ ("box") est utilisé. Au départ, $\Box\Phi$ signifiait "il est nécessaire que $\Phi$", et $\neg\Box\neg\Phi$, aussi noté $\Diamond\Phi$, signifiait "il est possible que $\Phi$". Lewis a ensuite proposé plusieurs ensembles de règles pour les nouveaux connecteurs, et a discuté leurs significations intuitives possibles.

Certaines de ces interprétations relèvent de réflexions philosophiques sur la notion de conséquence et d'implication. Considérer par exemple la formule $(\Box\Phi) \Rightarrow \Phi$. Si nous interprétons $\Box\Phi$ comme "A sait que $\Phi$ est vrai", alors $\Box\Phi \Rightarrow \Phi$ signifie "Si A sait que $\Phi$ est vrai, alors $\Phi$ est vrai"; si ceci est le cas pour tout $\Phi$, cette formule affirme que A ne fait pas d'erreur. La logique qui en résulte est appelée une logique de la connaissance. Ces logiques peuvent aussi modéliser le raisonnement temporel : si $\Box\Phi$ signifie "dans tous les futurs possibles, $\Phi$" (ou "à partir de maintenant, $\Phi$"), alors $(\Box\Phi) \Rightarrow \Phi$ dit que chaque fois que $\Phi$ est vraie à partir d'un certain instant, alors elle est vrai à cet instant précis aussi.

Dans ces interprétations, la formule $\Phi \Rightarrow \Box\Phi$ n'est pas vraie en général. Par exemple, dans l'interprétation temporelle, $\Phi \Rightarrow \Box\Phi$ signifie que si $\Phi$ est vrai à un instant, alors elle restera toujours vraie après coup.

D'un autre côté, si nous avons prouvé $\Phi$ dans un système modal, nous pourrons en déduire $\Box\Phi$. Intuitivement, si $\Phi$ est prouvée, alors elle est *toujours* vraie. (C'est la raison pour laquelle $\Box$ se lit "nécessairement".) Autrement dit, si $\Phi$ est prouvée, alors elle est vraie à tout instant, donc dans tous les futurs de tous les instants. Le fait surprenant que l'on puisse déduire $\Box\Phi$ de $\Phi$, mais que $\Phi \Rightarrow \Box\Phi$ ne soit pas valide veut dire que l'implication $\Rightarrow$ et la conséquence $\vdash$ ne coïncideront pas dans les logiques modales; autrement dit, le théorème de la déduction ne sera pas vérifié.

Pourquoi nous intéressons-nous à de telles logiques ? Laissant de côté les considérations philosophiques, les logiques modales et en particulier les logiques temporelles ont fait leur preuve comme des langages suffisamment expressifs, tout en restant décidables, pour exprimer des spécifications de modèles de calcul séquentiels et parallèles, et grâce auxquels la vérification d'une réalisation informatique par rapport à une spécification est décidable par des techniques simples à base d'automates. D'un point de vue logique, il est aussi intéressant d'explorer les rapports entre les logiques modales et non modales (comme la logique intuitionniste), pour mieux comprendre ce qui les rapprochent. Finalement, les logiques modales ont été utilisées dans la formalisation des *logiques non monotones*, dont le but est de modéliser des affirmations comme "en général, $\Phi$ est vrai" qui puissent survivre à la découverte de cas particuliers où $\Phi$ ne serait pas vraie.

Nous étudions maintenant ces aspects des logiques modales. Nous étudions d'abord la logique modale S4 en section 2, et illustrons sur cet exemple la formalisation des logiques non monotones. Nous choisissons S4 parce qu'elle est une des logiques modales les plus simples et les plus intéressantes, et aussi parce que c'est une logique temporelle. Ceci nous ménera naturellement à l'étude de logiques temporelles servant à la spécification de propriétés de systèmes de transitions en section 3, et de méthodes de vérification qu'un système de transitions satisfait une spécification dans ces logiques, la *vérification de modèle* (en anglais, "model-checking").

Le lecteur qui s'intéresse aux aspects logiques des logiques modales est prié de consulter [Che80] ou [HC68, HC84]. [Eme90, KT90] sont des textes sur les logiques temporelles et la vérification de modèle.

## 2 S4 et logiques non monotones

### 2.1 Syntax

Les formules de S4 sont toutes les formules construites à partir de variables propositionnelles à l'aide des connecteurs propositionnels usuels $\wedge$, $\vee$, $\neg$, $\Rightarrow$, et la *modalité* $\Box$ : si $\Phi$ est une formule, alors $\Box\Phi$ est une formule aussi ("box $\Phi$", ou "nécessairement, $\Phi$; en termes temporels, "dans tout futur, $\Phi$). Nous supposons que $\Box$ a une priorité plus forte que tous les autres opérateurs logiques, de sorte que $\Box A \Rightarrow A$ est une abréviation de $(\Box A) \Rightarrow A$, par exemple. Nous définissons aussi $\Diamond\Phi$ comme une abréviation de $\neg\Box\neg\Phi$ ("possiblement, $A$"; en termes temporels, "à un moment dans un futur, $A$").

En plus des règles de preuve de la logique classique, en S4 nous ajoutons les axiomes suivants :

(K) $\Box(\Phi \Rightarrow \Phi') \Rightarrow \Box\Phi \Rightarrow \Box\Phi'$

(T) $\Box\Phi \Rightarrow \Phi$

(4) $\Box\Phi \Rightarrow \Box\Box\Phi$

pour chaque formule $\Phi$ et $\Phi'$, et en plus de la règle de modus ponens (MP) rule, nous avons la règle de *nécessitation* :

(Nec) de $\Phi$, déduire $\Box\Phi$, pour chaque formule $\Phi$.

Les axiomes (K) et (T) sont communs à la plupart des logiques modales. L'axiome (K) dit que si $\Phi \Rightarrow \Phi'$ et $\Phi$ sont nécessaires (ou vraies dans tout futur), alors $\Phi'$ est nécessaire (ou vraie dans tout futur, dans l'interprétation temporelle). L'axiome (T) et la règle (Nec) ont été commentées au début de ce chapitre.

L'ensemble des axiomes (K) et (T) est aussi appelé (S). La logique S4 est en fait nommée d'après ses axiomes, et est en fait parfois appelée KT4.

Ce qui fait de S4 une logique particulière est principalement l'axiome (4). Dans l'interprétation de $\Box$ comme un opérateur de futur, (4) exprime la transitivité du temps, c'est-à-dire que si $\Phi$ est vrai dans tous les futurs, alors elle est aussi vraie dans tous les futurs de tous ces futurs.

Comme nous l'avons remarqué précédemment, les systèmes de Hilbert ne sont pas faciles à manipuler, et nous allons voir la sémantique de S4 en section 2.3, et des systèmes de séquents en section 2.5.

## 2.2 Logique philosophique et logiques non monotones

Il existe une multitude d'autres logiques modales venant de la communauté de logique philosophique. Toutes ont la règle (Nec); la plupart sont fondées sur l'axiome (K), et sont alors appelées logiques modales *normales*. Certaines utilisent l'axiome (T). D'autres axiomes utilisés pour définir des logiques modales sont :

(5) $\Diamond\Phi \Rightarrow \Box\Diamond\Phi$,

(D) $\Box\Phi \Rightarrow \Diamond\Phi$,

(R) $\Diamond\Box\Phi \Rightarrow (\Phi \Rightarrow \Box\Phi)$,

(F) $(\Box\Phi \Rightarrow \Phi') \vee (\Diamond\Box\Phi' \Rightarrow \Phi)$,

(2) $\Diamond\Box\Phi \Rightarrow \Box\Diamond\Phi$

(T) est parfois nommé (M), (5) parfois (E) (pour "euclidien"), et (R) est parfois appelé (W5).

Nous n'examinerons pas ce que les diverses combinaisons de ces axiomes signifient : voir le livre de Chellas.

Dans toute logique modale $L$, on peut définir une notion de déduction $\vdash^L$ : si $\Gamma$ est un ensemble de formules, et $\Phi$ est une formule, $\Gamma \vdash^L \Phi$ si et seulement si $\Phi$ peut être déduit de l'ensemble d'hypothèses $\Gamma$ et des axiomes de $L$ en utilisant uniquement les règles (MP) et (Nec). Mais, comme nous l'avons déjà dit, le théorème de la déduction n'est pas valide dans les logiques modales, autrement dit $\Phi_1, \ldots, \Phi_n \vdash^L \Phi$ n'est pas équivalent à $\vdash^L \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_n \Rightarrow \Phi$ (bien que ce dernier implique le premier).

Cette définition de la déductibilité est adaptée de celle de Rajeev Goré [Gor91], et est appelée *déductibilité forte*. La *déductibilité faible* est la relation $\vdash_L$ définie par $\Gamma \vdash_L \Phi$ si et seulement s'il existe un sous-ensemble fini $\{\Phi_1, \ldots, \Phi_n\}$ de $\Gamma$ tel que $\vdash^L \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_n \Rightarrow \Phi$. (Remarquer que, de façon paradoxale, la déductibilité faible implique la déductibilité forte, mais pas l'inverse.)

Les déductibilités faible et forte coïncident lorsque l'ensemble d'hypothèses $\Gamma$ est vide, et une formule $\Phi$ telle que $\vdash^L \Phi$ (autrement dit, $\vdash_L \Phi$) est appelée un *théorème* de $L$.

Un des grands intérêts des logiques modales est que la déductibilité forte plus un certain processus de complétion forment l'essence des *logiques non monotones*, qu'elles soient modales ou non. En traduisant Rajeev Goré [Gor91] :

> Les logiques modales propositionnelles sont utilisées pour modéliser des notions épistémiques comme la connaissance et la croyance depuis quelque temps, où la formule $\Box A$ se lit "on croit que $A$" ou "on sait $A$". Étant donnée une logique modale **S** (monotone) et un ensemble de formules $\Gamma$, la formule $A$ est une conséquence monotone de $\Gamma$ dans **S** si elle est déductible dans **S** à partir de $\Gamma$, ce que l'on écrit usuellement $\Gamma \vdash_S A$. L'ensemble $\Gamma$ est habituellement appelé une théorie et les conséquences monotones de $\Gamma$ dans **S** sont les formules déductibles de $\Gamma$ dans **S**; c'est-à-dire $Cn_S(\Gamma) = \{A \mid \Gamma \vdash_S A\}$. Le système est "monotone" en ce sens que si $A$ est dans $Cn_S(\Gamma)$, alors elle sera dans $Cn_S(\Gamma')$ pour tout sur-ensemble $\Gamma'$ de $\Gamma$.
>
> Pour obtenir la non-monotonie nous faisons l'hypothèse que $\neg\Box A$ ("on ne sait pas $A$") s'il n'y a pas de déduction de $A$ dans **S** à partir de $\Gamma$ et des hypothèses précédentes. Plus formellement, la théorie $T$ est une **S**-expansion de la théorie $\Gamma$ si elle satisfait l'équation $T = Cn_S(\Gamma \cup \{\neg\Box A \mid A \notin T\})$. Comme $T$ apparaît sur le côté droit, la définition est circulaire, et conséquemment, une théorie $\Gamma$ peut avoir zéro, une ou plusieurs **S**-expansions. Pour contre-balancer ce phénomène, l'ensemble des conséquences *non monotones* de $\Gamma$ dans **S** est usuellement défini comme l'intersection de toutes les **S**-expansions ed $\Gamma$. Le nouveau système est "non-monotone" parce que, bien que $A$ puisse être une conséquence non-monotone de $\Gamma$, elle peut ne pas être une conséquence non-monotone d'un sur-ensemble de $\Gamma$.

L'intérêt porté aux logiques non monotones apparaît lorsqu'on cherche à capturer la notion de raisonnement de tous les jours, où des conclusions par défaut sont obtenues à partir d'informations incomplètes, pour être éventuellement rétractées lors de l'apparition de contre-exemples. La *circonscription* de John McCarthy a été la première tentative de formalisation du raisonnement de tous les jours, suivie par la *logique des cas par défaut* de Reiter (*default logic*) et de la *logique auto-épistémique* de Halpern et Moses. Marek, Schwarz et Truszczyński ont ensuite remarqué que la logique auto-épistémique n'était rien d'autre que K45D non monotone, et que S4F généralisait aussi bien la logique des cas pas défaut que la logique auto-épistémique.

## 2.3 Sémantique de Kripke

La sémantique des logiques modales comme S4 sont usuellement décrites au moyen d'*interprétations en termes de mondes possibles* (*possible worlds interpretations*). En effet, une formule n'est pas juste vraie dans une interprétation, mais aussi dans un certain état, par exemple à un instant donné. Nous représentons l'ensemble de tous les états (par exemple, de tous les instants) par un *univers* $W$ non vide, dont les points, les états, sont appelés les *mondes*. Dans le cas de S4, les mondes sont partiellement ordonnés : si $w$ et $w'$ sont des mondes (des instants), $w \leq w'$ signifie que $w$ est un instant qui est avant $w'$. Alors une formule $\Box\Phi$ est vraie dans un monde $w$ si et seulement si $\Phi$ est vraie dans tous les mondes qui le suivent (à tous les instants ultérieurs).

La définition se généralise à toutes les logiques modales de la façon suivante, due à Saul Kripke in 1963 :

**Définition 1** *Un* cadre de Kripke *(*Kripke frame*) est un couple $(W, R)$, où $W$ est un ensemble non vide appelé l'*univers *et $R$ est une relation binaire sur $W$ appelée la* relation d'accessibilité*. Les éléments de $W$ sont appelés les* mondes.

*Une* affectation *ou* interprétation *$\rho$ est une fonction des variables propositionnelles vers les ensembles de mondes.*

*La sémantique d'une formule modale $\Phi$ est définie par la fonction à valeurs booléennes $[\![\Phi]\!]w\rho$, où $w$ est un monde et $\rho$ est une affectation. Nous notons $w, \rho \models \Phi$ pour $[\![\Phi]\!]w\rho$, et définissons :*

- *$w, \rho \models A$ si et seulement si $w \in \rho(A)$, où $A$ est une variable propositionnelle;*
- *$w, \rho \models \Phi \wedge \Phi'$ si et seulement si $w, \rho \models \Phi$ et $w, \rho \models \Phi'$;*
- *$w, \rho \models \Phi \vee \Phi'$ si et seulement si $w, \rho \models \Phi$ ou $w, \rho \models \Phi'$;*
- *$w, \rho \models \neg\Phi$ si et seulement si $w, \rho \not\models \Phi$;*
- *$w, \rho \models \Phi \Rightarrow \Phi'$ si et seulement si $w, \rho \not\models \Phi$ ou $w, \rho \models \Phi'$;*
- *$w, \rho \models \Box\Phi$ si et seulement si pour tout monde $w'$ tel que $wRw'$, alors $w' \models \Phi$.*

*Lorsque $w, \rho \models \Phi$, nous disons que $w, \rho$* satisfait *$\Phi$.*

*Un cadre de Kripke est* satisfait *par $\Phi$, ou est un* modèle de *$\Phi$ si et seulement si $w, \rho \models \Phi$ pour un certain monde $w$ et une certaine affectation $\rho$. $\Phi$ est* valide *dans le cadre donné si et seulement si $w, \rho \models \Phi$ pour tout monde $w$ et toute affectation $\rho$.*

*Si $\mathcal{C}$ est une classe de cadres, nous disons que $\Phi$ est* valide *dans $\mathcal{C}$, ce que nous notons $\mathcal{C} \models \Phi$, si et seulement si $\Phi$ est valide dans tout cadre de la classe.*

Remarquer que, si l'on regarde $R$ comme un ensemble de couples, un cadre de Kripke $(W, R)$ n'est rien d'autre qu'un *graphe orienté* de sommets les éléments de $W$ et d'arcs les éléments de $R$.

Il s'avère qu'une formule de S4 est un théorème si et seulement si elle est valide dans la classe des ensembles pré-ordonnés, c'est-à-dire des cadres $(W, R)$ tels que $R$ est une relation réflexive et transitive sur $W$. Si l'on voit les cadres comme des graphes, ceci signifie que les théorèmes de S4 sont en fait des théorèmes portant sur l'ensemble des chemins dans les collections de graphes orientés acycliques (en fait, les arbres suffisent). Plus précisément :

**Définition 2** *Soit $\mathcal{C}$ une classe de cadres, et $L$ une logique modale.*

*$L$ est* correcte *pour $\mathcal{C}$ si et seulement si pour toute formule $\Phi$, $\vdash^L \Phi$ implique $\mathcal{C} \models \Phi$.*

*$L$ est* complète *par rapport à $\mathcal{C}$ si et seulement si pour toute formule $\Phi$, $\mathcal{C} \models \Phi$ implique $\vdash^L \Phi$.*

On caractérise les cadres par les propriétés de leur relation d'accessibilité $R$ : elles peuvent être réflexives, symétriques, transitives, séquentielles ("serial" en anglais : depuis chaque monde il existe un monde accessible), convergentes (pour tout monde $w$, si $w_1$ et $w_2$ sont accessibles depuis $w$, alors il y a un monde accessible depuis $w_1$ et $w_2$ à la fois).

Alors, dans le cas de S4 :

**Théorème 3 (Correction, Complétude)** *S4 est correcte et complète par rapport à la classe des cadres réflexifs et transitifs.*

**Preuve :** Que S4 soit correcte pour les cadres réflexifs et transitifs est clair.

Pour prouver la complétude, et au lieu de montrer que $\models \Phi$ implique $\vdash^{S4} \Phi$ (ou de façon équivalente, $\vdash_{S4} \Phi$), nous allons supposer que $\not\vdash_{S4} \Phi$, et montrer que $\not\models \Phi$.

Choisissons une formule $\mathbf{F}$ représentant le faux, c'est-à-dire une formule qui implique toutes les autres prouvablement. Appelons un ensemble de formules $\Gamma$ *cohérent* s'il n'est pas le cas que $\Gamma \vdash_{S4} \mathbf{F}$. (De façon équivalente, tel que $\mathbf{F}$ n'est pas faiblement déductible de $\Gamma$.) Si $\not\vdash_{S4} \Phi$, alors $\neg\Phi$ est cohérent. En effet, si nous avions $\neg\Phi \vdash_{S4} \mathbf{F}$, alors nous aurions $\vdash_{S4} \neg\Phi \Rightarrow \mathbf{F}$, donc $\vdash_{S4} \Phi$ par raisonnement classique (c'est-à-dire sans utiliser aucune règle et aucun axiome modaux).

Nous construisons maintenant un modèle de $\neg\Phi$ (un cadre et une interprétation tels que $\neg\Phi$ soit vraie en un certain monde). Définissons pour cela le cadre de Kripke $(W, \leq)$, où $W$ est l'ensemble de tous les ensembles cohérents maximaux de propositions de S4, et où $w \leq w'$ si et seulement si pout toute formule de la forme $\Box\Phi'$ dans $w$, $\Phi'$ est dans $w'$. Définissons enfin l'interprétation $\rho$ par : pour chaque variable $A$, $\rho(A)$ est l'ensemble des mondes $w$ tels que $A \in w$.

En premier, faisons quelques remarques sur les ensembles cohérents maximaux $w$. Comme $w$ est cohérent, il ne peut pas contenir à la fois $\Phi$ et $\neg\Phi$, pour aucune formule $\Phi$. Mais pour toute formule $\Phi$, $w$ doit contenir soit $\Phi$ soit $\neg\Phi$. En effet, supposons que $w$ ne contienne ni $\Phi$ ni $\neg\Phi$. Alors $w \cup \{\Phi\}$ et $w \cup \{\neg\Phi\}$ sont tous les deux incohérents par maximalité de $w$, donc il existe un ensemble fini de formules $\Phi_1, \ldots, \Phi_n$ dans $w \cup \{\Phi\}$ tel que $\vdash^{S4} \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_n \Rightarrow \mathbf{F}$, et de même il existe un ensemble fini de formules $\Phi'_1, \ldots, \Phi'_{n'}$ dans $w \cup \{\neg\Phi\}$ tel que $\vdash^{S4} \Phi'_1 \Rightarrow \ldots \Rightarrow \Phi'_{n'} \Rightarrow \mathbf{F}$. Sans perte de généralité, supposons que $\Phi$ fait partie des $\Phi_i$ (sinon nous l'y ajoutons), et en fait que $\Phi = \Phi_n$, et de même nous pouvons supposer que $\neg\Phi = \Phi'_{n'}$. Nous pouvons aussi supposer, en ajoutant suffisamment de formules à chacun de ces deux ensembles, que les ensembles $\{\Phi_1, \ldots, \Phi_{n-1}\}$ et $\{\Phi'_1, \ldots, \Phi'_{n'-1}\}$ coïncident. Alors on a $\vdash^{S4} \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_{n-1} \Rightarrow \Phi \Rightarrow \mathbf{F}$ et $\vdash^{S4} \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_{n-1} \Rightarrow \neg\Phi \Rightarrow \mathbf{F}$, donc $\vdash^{S4} \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_{n-1} \Rightarrow (\Phi \vee \neg\Phi) \Rightarrow \mathbf{F}$, mais ceci revient à dire que $\vdash^{S4} \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_{n-1} \Rightarrow \mathbf{F}$. De plus $\Phi_1, \ldots, \Phi_{n-1}$ sont toutes dans $w$, donc $w$ est incohérent, ce qui est contradictoire. Donc $w$ doit contenir soit $\Phi$ soit $\neg\Phi$.

Observer aussi que tout ensemble cohérent maximal est stable par déduction faible : si $w$ est un ensemble cohérent maximal tel que $w \vdash_{S4} \Phi$, et si $\neg\Phi$ était dans $w$, nous pourrions déduire $\mathbf{F}$ de $w$ en déduisant d'abord $\Phi$, et en utilisant $\neg\Phi$ ensuite. Donc $\neg\Phi \notin w$, donc $\Phi \in w$.

Enfin, nous faisons la remarque que tout ensemble cohérent est inclus dans un ensemble cohérent maximal. Pour le montrer, nous utilisons le lemme de Zorn (une autre forme de l'axiome du choix), qui énonce que tout ensemble ordonné inductif $S$ a un élément maximal. Rappelons qu'un ensemble est *inductif* si et seulement si tout sous-ensemble totalement ordonné a une borne supérieure. Dans notre cas, soit $S$ l'ensemble de tous les ensembles cohérents de formules, ordonnés par inclusion. Étant donné un ensemble totalement ordonné d'ensemble cohérents $C_i$, $i \in I$, soit $C = \bigcup_{i \in I} C_i$. $C$ est cohérent, car sinon il existerait un nombre fini de formules $\Phi_1, \ldots, \Phi_n$ dans $C$ telles que $\vdash^{S4} \Phi_1 \Rightarrow \ldots \Rightarrow \Phi_n \Rightarrow \mathbf{F}$. Chaque $\Phi_i$ est dans un certain $C_i$, et comme les $C_i$ sont totalement ordonnés, il existe un $C_i$ contenant toutes les formules $\Phi_1, \ldots, \Phi_n$; mais alors $C_i$ serait incohérent, ce qui est contradictoire. Donc $S$ est inductif et par le lemme de Zorn, on peut compléter tout ensemble cohérent pour obtenir un ensemble cohérent maximal.

Nous affirmons en premier que $(W, \leq)$ est un cadre réflexif et transitif, c'est-à-dire que $\leq$ est un pré-ordre. Il est en effet réflexif, parce que pour tout $\Box\Phi' \in w$, $\Phi'$ is faiblement déductible de $\Box\Phi'$ par modus ponens avec (T); comme tout ensemble cohérent maximal est clos par déductibilité faible, $\Phi' \in w$. La relation $\leq$ est aussi transitive : si $w_1 \leq w_2 \leq w_3$, alors pour toute formule $\Box\Phi' \in w_1$, $\Box\Box\Phi' \in w_1$ by (4), donc $\Box\Phi' \in w_2$ puisque $w_1 \leq w_2$, donc $\Phi' \in w_3$ since $w_2 \leq w_3$; ceci entraîne $w_1 \leq w_3$.

Nous affirmons en second que pour toute formule $\Phi'$, $w, \rho \models \Phi'$ si et seulement si $\Phi' \in w$. Nous le montrons par récurrence structurelle sur $\Phi'$ :

- Si $\Phi'$ est une variable, c'est une conséquence de la définition de $\rho$.

- Si $\Phi'$ est de la forme $\Phi'' \vee \Phi'''$, alors : si $w, \rho \models \Phi'$, alors $w, \rho \models \Phi''$ ou $w, \rho \models \Phi'''$, donc $\Phi''$ ou $\Phi'''$ est dans $w$, par hypothèse de récurrence; mais si $\Phi'' \in w$ (resp. $\Phi''' \in w$), alors $\Phi' = \Phi'' \vee \Phi'''$ est faiblement déductible de $w$, donc $\Phi' \in w$.

  Réciproquement, si $w, \rho \not\models \Phi'$, alors $w, \rho \not\models \Phi''$ et $w, \rho \not\models \Phi'''$, donc $\Phi'' \notin w$ et $\Phi''' \notin w$ par hypothèse de récurrence; donc, par maximalité de $w$, $\neg\Phi''$ et $\neg\Phi'''$ doivent être toutes les deux dans $w$, et donc les formules faiblement déductibles $\neg\Phi'' \wedge \neg\Phi'''$ et $\neg(\Phi'' \vee \Phi''')$ (autrement dit $\neg\Phi'$) sont dans $w$, et ainsi $\Phi'$ ne peut pas être dans $w$, par cohérence.

- Le cas des conjonctions est similaire.

- Si $\Phi'$ est de la forme $\neg\Phi''$, alors : si $w, \rho \models \Phi'$, alors $w, \rho \not\models \Phi''$, donc par hypothèse de récurrence $\Phi'' \notin w$; par maximalité de $w$, $w \cup \{\Phi''\}$ n'est pas cohérent, en particulier $\neg\Phi''$ ne peut pas être faiblement déductible de $w \cup \{\Phi''\}$, et donc de $w$; en conséquence $\Phi' = \neg\Phi'' \in w$. Réciproquement, si $\Phi' \in w$, alors $\Phi''$ n'est pas dans $w$ car sinon $w$ ne serait pas cohérent, donc par hypothèse de récurrence $w, \rho \not\models \Phi''$, ce qui implique $w, \rho \models \Phi'$.

- Si $\Phi'$ est de la forme $\Box\Phi''$, alors : si $\Phi' \in w$, alors $\Phi'' \in w'$ pour tout $w' \geq w$ par définition de $\leq$, donc par hypothèse de récurrence $w', \rho \models \Phi''$ pour tout $w' \geq w$, ce qui implique $w, \rho \models \Phi'$.

  Réciproquement, si $w, \rho \models \Phi'$, alors $w', \rho \models \Phi''$ pour tout $w' \geq w$, donc par hypothèse de récurrence $\Phi'' \in w'$ pour tout $w' \geq w$. Maintenant, considérons l'ensemble $w'_0 = \{\Psi \mid \Box\Psi \in w\}$.

  Nous affirmons que $w'_0 \cup \{\neg\Phi''\}$ est incohérent. En effet, supposons le contraire, autrement dit supposons que $w'_0 \cup \{\neg\Phi''\}$ est cohérent. Complétons-le pour former un sur-ensemble cohérent maximal $w'_1$. Par définition de $w'_0$, de $w'_1$ et de $\leq$, $w \leq w'_1$. Comme $\Phi'' \in w'$ pour tout $w' \geq w$, nous devons donc avoir $\Phi'' \in w'_1$. Mais alors $w'_1$ contient à la fois $\Phi''$ et $\neg\Phi''$ (par construction), et est donc incohérent.

  Donc $w'_0 \cup \{\neg\Phi''\}$ est incohérent, et nous pouvons faiblement déduire $\mathbf{F}$ de $w'_0, \neg\Phi''$, donc par raisonnement classique nous pouvons dériver $\Phi''$ de $w'_0$. (Observe que la déduction faible satisfait le théorème de la déduction par définition.)

Mais alors, il existe un nombre fini de formules $\Psi_1, \ldots, \Psi_n$ dans $w'_0$ telles que $\vdash_{S4} \Psi_1 \Rightarrow \ldots \Rightarrow \Psi_n \Rightarrow \Phi''$. Donc $\vdash_{S4} \Box(\Psi_1 \Rightarrow \ldots \Rightarrow \Psi_n \Rightarrow \Phi'')$ par (Nec), donc $\vdash_{S4} \Box\Psi_1 \Rightarrow \ldots \Rightarrow \Box\Psi_n \Rightarrow \Box\Phi''$ par $n$ applications de l'axiome (K) et le raisonnement classique. Il s'ensuit que $\Box\Phi''$, c'est-à-dire $\Phi'$, est faiblement déductible à partir de $\Box\Psi_i$, $1 \leq i \leq n$, qui sont toutes dans $w$ par construction. Donc $\Phi'$ est elle-même dans $w$.

Finalement, comme $\neg\Phi$ est cohérent, on peut compléter $\{\neg\Phi\}$ pour former un ensemble cohérent maximal $w$, et par définition de $W$, $\leq$ et $\rho$, ceci implique que $w, \rho \models \neg\Phi$. □

Ce théorème signifie que $\Phi$ est prouvable en S4 si et seulement si elle est vraie dans tout cadre dont la relation d'accessibilité est un pré-ordre. En fait, on peut se restreindre aux cadres où la relation d'accessbilité est un ordre, c'est-à-dire où elle est aussi anti-symétrique. (C'est parce que les modèles définis par filtration ci-dessous seront anti-symétriques.)

Bien que nous n'ayons prouvé le théorème que pour S4, des théorèmes analogues peuvent être prouvés pour les autres logiques. Par exemple, KD est correct et complet pour tous les cadres séquentiels, S pour les cadres réflexifs, K4 pour les cadres transitifs, S5 pour toutes les relations d'équivalence (réflexives, symétriques, transitives), et S4.2 pour tous les pré-ordres convergents. La preuve est exactement la même que ci-dessus. En particulier, nous construisons la relation d'accessibilité $\leq$ en disant que $w \leq w'$ si et seulement si pour toute formule $\Box\Psi$ dans $w$, $\Psi$ est dans $w'$. La seule différence est que $\leq$ sera quelque chose d'autre : une relation séquentielle pour KD, une relation réflexive pour S, etc.

## 2.4 Décidabilité

Pour nous aider à montrer qu'une logique modale $L$ est décidable, définissons une relation d'équivalence sur les mondes (une relation d'indistingabilité) et le quotient des cadres par cette relation. Cette définition est justifiée par la construction du théorème 7.

**Définition 4 (Filtration)** *Soit $\Phi$ une formule de $L$, et $\rho$ une affectation. Soit $S(\Phi)$ l'ensemble des sous-formules de $\Phi$.*

*La* relation d'indistingabilité *$\equiv_{\Phi,\rho}$ sur un cadre $(W, R)$ est définie par $w \equiv_{\Phi,\rho} w'$ si et seulement si pour toute formule $\Phi' \in S(\Phi)$, $w, \rho \models \Phi'$ si et seulement si $w', \rho \models \Phi'$.*

*La* filtration *de $(W, R)$ par $S(\Phi), \rho$ est le cadre $(W/\Phi\rho, R/\Phi\rho)$, où $W/\Phi\rho$ est le quotient de $W$ par $\equiv_{\Phi,\rho}$ et si $v$, $v'$ sont deux éléments de $W/\Phi\rho$, alors $v(R/\Phi\rho)v'$ si et seulement s'il existe $w \in v$, $w' \in v'$ tels que $wRw'$.*

*Soit $w_{\Phi,\rho}$ la classe de $w$ modulo $\equiv_{\Phi,\rho}$, et soit $\rho_\Phi$ l'affectation dans $(W/\Phi\rho, R/\Phi\rho)$ qui envoie chaque variable $A$ vers le quotient $\rho(A)/\Phi\rho = \{w_{\Phi,\rho} \mid w \in \rho(A)\}$.*

La relation $\equiv_\Phi$ est en effet une relation d'équivalence sur les mondes. La filtration préserve les modèles :

**Lemme 5** *Pour toute sous-formule $\Phi'$ de $\Phi$, $(W, R, \rho) \models \Phi'$ si et seulement si $(W/\Phi\rho, R/\Phi\rho, \rho_\Phi) \models \Phi'$.*

**Preuve :** Nous affirmons que : (1) si $v \in W/\Phi\rho$, alors $v, \rho_\Phi \models \Phi'$ si et seulement si l'une des deux conditions équivalentes suivantes est vérifiée :

1. il y a un $w \in W$ tel que $v = w_{\Phi,\rho}$ et $w, \rho \models \Phi'$;
2. pour tout $w \in W$ tel que $v = w_{\Phi,\rho}$, alors $w, \rho \models \Phi'$.

Ces conditions sont équivalentes; en effet, 2 implique 1 parce que les classes d'équivalence sont non vides, et 1 implique 2 parce que les mondes d'une même classe d'équivalence sont indistingables et parce que $\Phi'$ est une sous-formule de $\Phi$. L'affirmation (1) est alors montrée par récurrence structurelle sur $\Phi'$.

En guise d'illustration, traitons le cas où $\Phi'$ est de la forme $\Box\Phi''$. Si $v, \rho_\Phi \models \Box\Phi''$, alors pour tout $v'$ tel que $v(R/\Phi\rho)v'$, $v', \rho_\Phi \models \Phi''$ est vrai; donc, pour tout $w \in v$, pour tout $w'$ tel que $w\ R\ w'$, $w'/\Phi\rho, \rho_\Phi \models \Phi''$ par définition de $R/\Phi\rho$; par hypothèse de récurrence, $w', \rho \models \Phi''$ pour tous ces $w'$, donc $w, \rho \models \Box\Phi''$. Réciproquement, si pour tout $w \in v$, $w, \rho \models \Box\Phi''$, alors pour tout $w'$ tel que $w\ R\ w'$, $w', \rho \models \Phi''$; donc pour tout $v'$ tel que $v(R/\Phi\rho)v'$, il y a un $w \in v$ et un $w' \in v'$ tels que $w\ R\ w'$, d'où $w', \rho \models \Phi''$, et par hypothèse de récurrence, $v', \rho_\Phi \models \Phi''$; comme $v'$ est arbitraire, il s'ensuite que $v,\ rho_\Phi \models \Box\Phi''$. □

De plus, la filtration préserve la plupart des propriétés du modèle :

**Lemme 6** *Si $(W, R)$ est réflexif (resp. symétrique, séquentiel), alors $(W/\Phi\rho, R/\Phi\rho)$ est réflexif (resp. symétrique, séquentiel).*

**Preuve :** Si $(W, R)$ est réflexif, alors pour tout $v \in W/\Phi\rho$, il existe un monde $w \in v$, et $w\ R\ w$ par hypothèse, donc $v(R/\Phi\rho)v$ par définition.

Si $(W, R)$ est symétrique, et $v(R/\Phi\rho)v'$, alors il existe $w, w' \in W$ tels que $w\ R\ w'$, donc $w'\ R\ w$ par symétrie, et donc $v'(R/\Phi\rho)v$.

Si $(W, R)$ est séquentiel, alors pour tout $v \in W/\Phi\rho$, il y a un $w$ dans $v$, et aussi un $w' \in W$ tel que $w\ R\ w'$ par hypothèse, donc $v(R/\Phi\rho)w'/\Phi\rho$, et $(W/\Phi\rho, R/\Phi\rho)$ est séquentiel. □

Cependant, quand $(W, R)$ est transitif, $(W/\Phi\rho, R/\Phi\rho)$ ne l'est pas nécessairement. En effet, si $v(R/\Phi\rho)v'$ et $v'(R/\Phi\rho)v''$, alors il existe des mondes $w$, $w'_1$, $w'_2$, $w''$ tels que $w\ R\ w'_1$ et $w'_2\ R\ w''$, mais on ne peut rien en conclure, car il n'y a pas nécessairement de relation entre $w'_1$ et $w'_2$. De même, quand $(W, R)$ est convergent, $(W/\Phi\rho, R/\Phi\rho)$ n'est pas nécessairement convergent. Nous réglerons ce problème bientôt.

Finalement, la filtration est suffisamment petite :

**Théorème 7 (Petit modèle)** *Soit $\Phi$ une formule de taille $n$. Le cardinal de $W/\Phi\rho$ est inférieur ou égal à $2^n$.*

**Preuve :** Pour tout $\Phi' \in S(\Phi)$, soit $f_0(\Phi') = \{w \in W \mid w, \rho \models \Phi'\}$, et $f_1(\Phi') = \{w \in W \mid w, \rho \not\models \Phi'\}$.

Maintenant, soient $\Phi_1, \ldots, \Phi_n$ les $n$ sous-formules de $\Phi$, et soit $s_1, \ldots, s_n$ une suite arbitraire d'éléments de $\{0, 1\}$. Alors $\bigcap_{i=1}^n f_{s_i}(\Phi_i)$ est une classe d'équivalence modulo $\equiv_{\Phi,\rho}$, c'est-à-dire un élément de $W/\Phi\rho$.

Réciproquement, toute classe d'équivalence est déterminée par la valeur booléenne (vrai si $s_i = 0$, faux si $s_i = 1$) de toutes les sous-formules $\Phi_i$, $1 \leq i \leq n$, de $\Phi$. Autrement dit toutes les classes d'équivalence sont de la forme $\bigcap_{i=1}^n f_{s_i}(\Phi_i)$ pour une certaine suite $s_1, \ldots, s_n$.

Il y a donc au plus autant de mondes dans $W/\Phi\rho$ qu'il n'y a de suites de longueur $n$ sur $\{0, 1\}$, soit au plus $2^n$. □

Ceci établit la propriété dite de *petit modèle* pour toutes les logiques pour lesquelles $(W/\Phi\rho, R/\Phi\rho, \rho_\Phi)$ est un modèle de $\Phi$ dès que $(W, R, \rho)$ en est un. Ceci fonctionne pour toute combinaison de réflexivité, de symétrie et de séquentialité.

Lorsque les modèles sont transitifs, $(W/\Phi\rho, R/\Phi\rho, \rho_\Phi)$ peut ne pas être un modèle de $\Phi$ du tout bien que $(W, R, \rho)$ en soit un. Dans ce cas, nous remplaçons le lemme 5 par :

**Lemme 8** *Supposons $R$ transitive.*

*Pour toute sous-formule $\Phi'$ de $\Phi$, $(W, R, \rho) \models \Phi'$ si et seulement si $(W/\Phi\rho, (R/\Phi\rho)^+, \rho_\Phi) \models \Phi'$, où $(R/\Phi\rho)^+$ est la clôture transitive de $R/\Phi\rho$.*

**Preuve :** Comme pour le lemme 5. □

Quant $R$ est transitive, donc, si $(W, R, \rho)$ est un modèle de $\Phi$, alors $(W/\Phi\rho, (R/\Phi\rho)^+, \rho_\Phi)$ est un autre modèle de $\Phi$, qui est clairement petit (fini). Une technique analogue fonctionne pour les cadres de Kripke convergents (resp. convergents et transitifs).

Ce qui fait que la propriété de petit modèle est utile est :

**Théorème 9 (Décidabilité)** *Soit $L$ une logique modale ayant la propriété de petit modèle. Alors la validité des formules $\Phi$ de $L$ est décidable.*

**Preuve :** Soit $n$ la taille de $\Phi$, et supposons que la taille d'un petit modèle pour $\Phi$ est majorée par $f(n)$. (Nous avions $f(n) = O(2^n)$ plus haut.) Énumérons tous les graphes ayant au plus $f(n)$ sommets et qui sont dans la classe $\mathcal{C}$. Pour ce faire, nous énumérons tous les graphes ayant exactement $m$ sommets, pour tout $1 \leq m \leq f(n)$ : il suffit d'énumérer tous les graphes sur $\{1, \ldots, m\}$, et il y en a au plus $2^{m^2}$. Sur un graphe à $m$ sommets, il y a au plus $2^{nm}$ affectations possibles (associant chaque variable et monde à un booléen disant si le variable est vrai à ce monde ou non), et nous vérifions chaque affectation en temps $O(nm)$ (calculant la valeur de vérité de chaque sous-formule en chaque monde, en partant des formules atomiques et en remontant dans $\Phi$). En somme, le temps d'exécution est majoré par $O(\sum_{m=1}^{f(n)} 2^{m^2}.2^{nm}.nm) \leq nf(n)^2.2^{f(n)^2+nf(n)}$. □

Ceci est le cas, comme nous l'avons vu, des classes de cadres satisfaisant n'importe quelle combinaison de réflexivité, de symétrie, de transitivité, de séquentialité et de convergence.

En jargon de théorie de la complexité, la borne supérieure du temps que prend l'algorithme de petit modèle ci-dessus est dite doublement exponentielle. Si l'on examine l'algorithme plus attentivement, nous voyons que déterminer si $\Phi$ est invalide demande de deviner un cadre et une affectation dans ce cadre en temps simplement exponentiel, et de les vérifier en temps simplement exponentiel. En somme, l'algorithme tourne en *temps non déterministe simplement exponentiel*. Cependant, la plupart des logiques modales sont décidables en *temps déterministe simplement exponentiel*, ce qui est beaucoup moins intractable, quoique toujours prouvablement intractable (autrement dit, certainement pas faisable en temps polynomial dans tous les cas). Le problème de la satisfiabilité dans la plupart de ces logiques, comme S4, est en fait PSPACE-complet, et pour S5 il n'est même "que" NP-complet.

L'astuce qui fait que la propriété de petit modèle fonctionne est de considérer une relation d'indistingabilité définie sur l'ensemble $S(\Phi)$ de toutes les sous-formules de $\Phi$. C'est une façon sémantique de dire que la sémantique de $\Phi$ ne dépend que de ses sous-formules, et c'est donc une version analogue de la propriété de sous-formule que l'on dérive usuellement comme une conséquence du processus syntaxique d'élimination des coupures.

## 2.5 Systèmes de séquents et tableaux

Melvin Fitting [Fit83] a été l'un des premiers logiciens à s'intéresser aux systèmes de preuve pour les logiques modales. Il s'avère que la plupart des méthodes de preuve connues pour la logique classique ne s'adaptent pas au logiques modales. Cependant, les méthodes de tableaux, à savoir les systèmes de séquents de Gentzen sans coupures, ont beaucoup de succès dans ce domaine. (Pour une autre méthode, voir [Ohl93].)

Mais l'élimination des coupures ou la propriété de sous-formule échoue dans certaines logiques modales. S4R [Gor91], par exemple, a un système de séquents sans coupures, mais qui n'a pas la propriété de sous-formule : on peut avoir besoin de sous-formules non seulement de $\Phi$, mais aussi de $\Box\Phi$ pendant la recherche de preuve. Ce n'est pas trop grave, puisque la décidabilité est conservée. Des logiques comme S4.2 et S4F ont des systèmes de Gentzen qui non seulement n'ont pas la propriété de sous-formule (on doit considérer toutes les sous-formules de $\Box\neg\Box\Phi$), mais exigent aussi des règles dites de *coupures analytiques*. Cette forme spéciale de la règle Cut est telle que la formule de coupure (celle qui disparaît dans la conclusion de Cut) est toujours une sous-formule directe de l'une des formules dans la conclusion de la règle Cut, ce qui limite l'explosion de l'espace de recherche; la décidabilité est encore préservée, mais à un prix élevé.

Examinons quelques exemples concrets. Un système de séquents dérivé de ce que Goré propose pour S4 est $\mathbf{LK}_0$ sans Cut plus les règles :

$$\frac{\Box\Gamma \longrightarrow \Phi}{\Box\Gamma, \Gamma' \longrightarrow \Box\Phi, \Delta}\,(\mathrm{S4}) \qquad \frac{\Gamma, \Phi \longrightarrow \Delta}{\Gamma, \Box\Phi \longrightarrow \Delta}\,(\mathrm{T})$$

où, si $\Gamma$ est un ensemble de formules, $\Box\Gamma$ est l'ensemble $\{\Box\Psi \mid \Psi \in \Gamma\}$.

Interprétons ces règles comme des règles de tableaux. Supposons que nous soyons sur un chemin non clos. Les formules de type $\alpha$ et $\beta$ s'expansent comme dans le cas classique. Mais si nous décidons d'expanser une formule de la forme $+\Box\Phi$, nous devons le faire en la remplaçant sur le chemin courant par $+\Phi$, et nous devons aussi masquer toutes les formules négatives $-\Psi$ telles que $\Psi$ n'est pas de la forme $\Box\Psi'$, ainsi que toutes les formules positives sauf $\Phi$ (règle (S4)). Et, si nous souhaitons expanser une formule de la forme $-\Box\Phi$, nous devons le faire en ajoutant $-\Phi$ sur le chemin. (La contraction implicite à gauche des séquents dans la règle $(T)$ n'est *pas* éliminable.)

Remarquer que, contrairement à la logique classique, nous ne pouvons pas choisir de décomposer n'importe quelle formule arbitraire sur le chemin : si nous décomposons une formule $+\Box\Phi$ en utilisant (S4), nous ne pourrons plus accéder aux formules de $\Gamma'$ ou $\Delta$, et il faudra que nous rebroussions chemin ("backtrack" en anglais), que nous choisissions une autre formule à décomposer si la première n'a mené à rien. À cause de règles comme (S4), les tableaux pour les logiques modales ne sont pas aussi simples que dans le cas classique.

D'autres logiques modales ont des systèmes de séquents beaucoup plus compliqués que S4, pourtant. Par exemple, voici une variante du système de Goré's pour S4.2. En plus des règles de $\mathbf{LK}_0$ (sans Cut), des règles (T) et (S4) ci-dessus, nous incluons :

$$\frac{\Gamma, \Box\neg\Box\Phi \longrightarrow \Box\Phi, \Delta \quad \Gamma, \Box(\neg\Box\neg\Box\Phi)^* \longrightarrow \Box\Phi, \Delta}{\Gamma \longrightarrow \Box\Phi, \Delta}\,(\mathrm{S4.2})$$

où $(\ldots)^*$ est appelée une formule marquée, et la règle est restreinte au cas où $\Box\Phi$ n'est pas marquée dans la conclusion. (Les marques sont juste des annotations sur des occurrences de formules, et ne participent pas de la structure de la formule; leur rôle est de détecter les bouclages, c'est-à-dire les situations où une même règle devrait être appliquée deux fois à la même formule.) Pour trouver une preuve en S4.2, lorsque nous en arrivons à expanser une formule de la forme $+\Box\Phi$, nous avons le choix entre appliquer la règle (S4) ou la règle (K45); cette dernière enrichit le chemin courant, et produit deux chemins sur lesquels (S4) produira des séquents plus riches en information.

Mais la règle (S4.2) ne suffit pas à trouver toutes les preuves, et nous devons ajouter les règles de coupures analytiques suivantes :

$$\frac{\Gamma \longrightarrow \Phi, \Phi', \Delta \quad \Gamma, \Phi \longrightarrow \Phi', \Delta \quad \Gamma, \Phi' \longrightarrow \Phi, \Delta}{\Gamma \longrightarrow \Phi \wedge \Phi', \Delta}$$

$$\frac{\Gamma, \Phi \longrightarrow \Box\Phi, \Delta \quad \Gamma \longrightarrow \Phi, \Box\Phi, \Delta}{\Gamma \longrightarrow \Box\Phi, \Delta}$$

ainsi que les règles correspondantes pour $\vee$ et les autres conncteurs. (Goré n'utilise que $\neg$, $\wedge$ et $\Box$ comme connecteurs de base.) Ces règles rajoute beaucoup de non-déterminisme et de branchement dans la recherche de preuve : trouver des preuves dans ces logiques n'est pas facile.

## 3 Autres logiques modales utilisées en informatique

Un échantillon de logiques modales utilisées encore plus couramment en informatique consiste en la logique de Hennessy-Milner ou HML, qui a été inventé pour fournir un langage logique pour décrire les processus parallèles du langage CCS de Milner; en des logiques temporelles comme CTL, conçues pour fournir un langage plus riche de spécification de circuits matériels, ou la logique propositionnelle dynamique (PDL), qui est essentiellement la logique de Hoare (une logique servant à raisonner sur des programmes), présentée de façon plus succincte et plus élégante. Nous nous intéresserons principalement au $\mu$-calcul modal de Kozen, une logique plus générale que celles mentionnées ci-dessus. (Voir [Mil89] pour CCS et HML, [Eme90] pour les logiques temporelles, et [KT90] pour la logique dynamique.)

L'idée de départ dans ces logiques est qu'elles fournissent un langage suffisamment riche pour exprimer des propriétés utiles sur des automates (à savoir des machines à états finis, encore connues sous le nom de graphes états-transitions étiquetés). Un automate est un ensemble $S$ d'*états*, muni d'une relation de transition $R$ entre les états. (Nous laissons de côté les états initiaux et finaux dans cette définition.) Une *transition* est un couple $(s, s')$ d'états dans $R$, et on la suppose étiquetée par une lettre prise dans un alphabet fini $\mathcal{A}$. Nous notons $s \xrightarrow{a} s'$ la transition $(s, s')$ étiquetée par $a$. Alors nous pouvons considérer S4 ou des logiques temporelles plus compliquées comme des langages d'expression de propriétés sur des automates, vus comme des cadres de Kripke : par exemple, en S4, le fait que $\Box\Phi$ soit satisfaite á l'état $s$ signifierait que $\Phi$ est satisfiate à tous les états que nous pouvons atteindre depuis $s$ en suivant un nombre fini de transitions. (Les chemins que nous pouvons prendre représentent tous les futurs possibles de $s$.)

La logique deHennessy-Milner logic (HML) est une version à plusieurs agents de la logique modale de base K; au lieu de n'avoir qu'une seule modalité $\Box$ (resp. $\Diamond$), nous avons plusieurs modalités $[s]$ (resp. $\langle s\rangle$), correspondant chacune à un ensemble possible $s$ de transitions dans un automate représentant toutes les exécutions possibles d'un programme CCS. Les transitions dans ces systèmes sont censées représenter des actions atomiques individuelles, comme envoyer ou recevoir un message. Les transitions peuvent être composées en séquence ou en parallèle. De plus, les transitions peuvent se synchroniser, et c'est la façon dont des processus tournant en parallèle arrivent à communiquer : pour chaque transition $i$ (par exemple, "envoyer tel message"), il existe une transition correspondante distincte $\bar{i}$ ("recevoir tel message"), telle que de deux processus dont l'un est prêt à déclencher $i$ et l'autre $\bar{i}$, peuvent déclencher les deux en même temps et continuer l'exécution. L'exécution procède en déclenchant les transitions d'un état au suivant. Un état est souvent appelé un *agent* ou un *process*. Nous écrivons $A \xrightarrow{i} A'$ pour dire que l'agent $A$ peut se transformer en l'agent $A'$ en suivant la transition $i$.

La logique de Hennessy-Milner représente les comportements des processus par des formules modales. Le fait qu'un agent $A$ satisfasse $[s]\Phi$, où $s$ est un ensemble de transitions, signifie : "pour toute transition $A \xrightarrow{i} A'$ avec $i \in s$, $A'$ satisfait $\Phi$", et $\langle s \rangle \Phi$ (i.e., $\neg[s]\neg\Phi$) signifie : "il y a une transition $A \xrightarrow{i} A'$ avec $i \in s$ telle que $A'$ satisfait $\Phi$". Bien que ce ne soit pas nécessaire, nous nous restreignons maintenant à des ensembles *finis* de transitions, par souci de simplicité.

D'après les définitions, nous pouvons supposer que $[\{i_1, \ldots, i_n\}]\Phi$ est une abréviation de $[i_1]\Phi \wedge \ldots \wedge [i_n]\Phi$ (ou d'une formule valide quelconque $\mathbf{T}$ si $n = 0$), où $[i]\Phi$ signifie $[\{i\}]\Phi$. Alors les axiomes sont ceux de la logique propositionnelle classique plus :

(K) $[i](A \Rightarrow B) \Rightarrow [i]A \Rightarrow [i]B$,

avec une règle de nécessitation modifiée :

(Nec) de $\Phi$, déduire $[i]\Phi$, pour toute formule $\Phi$ et toute transition $i$.

Cette logique peut être rendue plus expressive en l'étendant par, par exemple, un opérateur de plus petit point fixe $\mu$ et un opérateur de plus grand point fixe $\nu$; nous obtenons le *$\mu$-calcul modal*.

L'idée est que, si $A$ est une variable propositionnelle qui n'apparaît que positivement dans $\Phi$ (en supposant que $\Rightarrow$ a été remplacé par $\neg$ et $\vee$, ceci signifie que $A$ est à l'intérieur d'un nombre pair de négations; c'est un critère syntaxique assurant que si $A \Rightarrow A'$ est vrai, alors $\Phi \Rightarrow \Phi[A'/A]$ est vrai aussi), alors $\mu A \cdot \Phi$ et $\nu A \cdot \Phi$ sont deux formules $\Phi'$ logiquement équivalentes à $\Phi[\Phi'/A]$ : ce sont des *points fixes* de la fonction $A \mapsto \Phi$, modulo l'équivalence logique. De plus, chaque fois que $\Phi'$ est une formule telle que $\Phi'$ est équivalente à $\Phi[\Phi'/A]$, alors $\mu A \cdot \Phi \Rightarrow \Phi'$ et $\Phi' \Rightarrow \nu A \cdot \Phi$ sont vraies. On a les axiomes et règles suivantes :

- $(\mu A \cdot \Phi) \Leftrightarrow \Phi[(\mu A \cdot \Phi)/A]$, $(\nu A \cdot \Phi) \Leftrightarrow \Phi[(\nu A \cdot \Phi)/A]$,
- $\dfrac{\Phi[\Phi'/A] \Rightarrow \Phi'}{(\mu A \cdot \Phi) \Rightarrow \Phi'} \qquad \dfrac{\Phi' \Rightarrow \Phi[\Phi'/A]}{\Phi' \Rightarrow (\nu A \cdot \Phi)}$

Intuitivement, $\mu A \cdot \Phi$ est le plus petit point fixe de la fonction $f$ qui à $A$ associe $\Phi$, c'est-à-dire la disjonction de toutes les formules $\underbrace{f(f(\ldots f(}_{n \text{ times}}\mathbf{F})\ldots)$, $n \in \mathbb{N}$, où $\mathbf{F}$ est une formule donnée représentant le faux. Réciproquement, $\nu A \cdot \Phi$ est intuitivement la conjonction de toutes les formules $\underbrace{f(f(\ldots f(}_{n \text{ times}}\mathbf{T})\ldots)$, $n \in \mathbb{N}$, et $\mathbf{T}$ est une formule représentant le vrai.

Les opérateurs $\mu$ et $\nu$ nous permettent d'exprimer des propriétés qui ne pouvaient pas être décrites en HML, parce que $\mu$ et $\nu$ incorporent une forme de récurrence que nous n'avions pas en HML. Par exemple, la vivacité ("liveness" en anglais) peut s'exprimer $\nu A \cdot \langle \mathcal{A} \rangle \mathbf{T} \wedge [\mathcal{A}]A$, où $\mathcal{A}$ est l'ensemble de toutes les lettres de transitions; ceci veut dire qu'à chaque étape de calcul, nous pouvons exécuter une transition, et quelle que soit la transition que nous choisissions, nous pouvons alors encore exécuter une transition, et ainsi de suite. Nous pouvons aussi écrire qu'un processus peut bloquer ("deadlock" en anglais) comme $\mu A \cdot [\mathcal{A}]\mathbf{F} \vee \langle \mathcal{A} \rangle A$. En général, si $\Phi$ ne dépend pas de $A$, $\mu A \cdot \Phi \vee \langle \mathcal{A} \rangle A$ exprime qu'il existe un chemin d'exécution tel que $\Phi$ finit par être satisfait sur ce chemin, alors que $\mu A \cdot \Phi \vee [\mathcal{A}]A$ dit que, sur tous les chemins d'exécution, $\Phi$ finit par être satisfait (autrement dit, $\Phi$ est *inévitable*). Une formule plus compliquée comme $\nu B \cdot (\mu A \cdot \Phi \vee [\mathcal{A}]A) \wedge \langle \mathcal{A} \rangle \mathbf{T} \wedge [\mathcal{A}]B$ dit que $\Phi$ est vraie infiniment souvent sur chaque chemin d'exécution. (La $\mu$-formule interne signifie que $\Phi$ finit par être vraie, donc la formule tout entière dit que $\Phi$ finit par être vraie en un état, et qu'alors nous pouvons suivre une transition, et quelle que soit celle que nous choisissions, $\Phi$ finira par redevenir vraie, et ainsi de suite.)

Parce qu'il incorpore une forme de récurrence, le $\mu$-calcul modal n'est pas compact (contrairement à la logique propositionnelle par exemple) :

**Théorème 10 (Non-compacité)** *Il existe un ensemble infini $\Gamma$ de formules du $\mu$-calcul modal, et une formule du $\mu$-calcul modal $\Phi$, tels que $\Gamma \models \Phi$, mais $\Gamma, \Delta \not\models \Phi$ pour tout sous-ensemble fini $\Delta$ de $\Gamma$.*

**Preuve :** Soit $\Gamma$ l'ensemble contenant $\mu B \cdot A \vee \langle a \rangle B$ (qui dit qu'il existe un monde accessible depuis le monde courant par un nombre fini de transitions $a$ et où $A$ est vrai), et $\neg A$, $\neg\langle a \rangle A$, $\neg\langle a \rangle\langle a \rangle A$, ..., et soit $\Phi$ n'importe quelle formule insatisfiable.

On a $\Gamma \models \Phi$, parce que $\Gamma$ n'est satisfaite dans aucun modèle.

Cependant, pour tout sous-ensemble fini $\Delta$ de $\Gamma$, on peut construire un modèle de $\Delta$ comme suit : soit $k$ un entier tel que pour tout $j \geq k$, $\neg \underbrace{\langle a \rangle \ldots \langle a \rangle}_{j \text{ times}} A$ n'est pas dans $\Delta$, et on construit le graphe orienté consistant en exactement un circuit

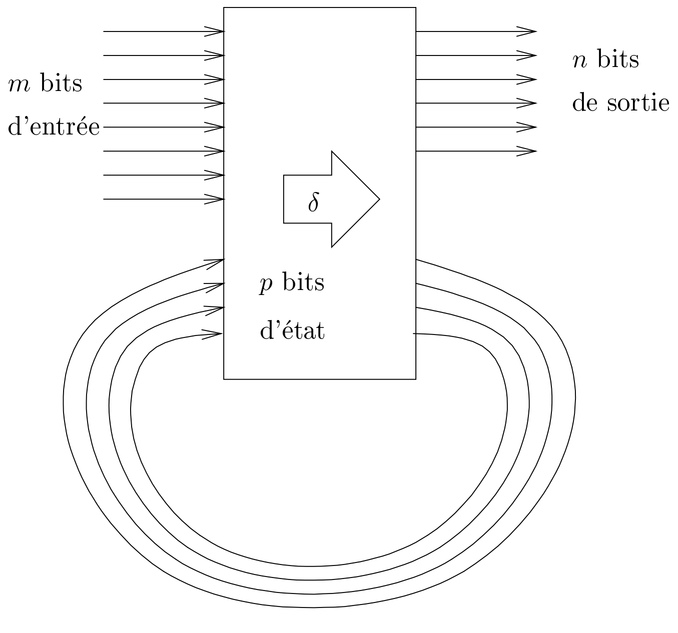


Figure 1: Une machine séquentielle

de $k+1$ états, numérotés de 0 à $k$; on définit $R_a$ comme la relation qui envoie chaque sommet du circuit vers son successeur (autrement dit, $i$ vers $i+1$ si $0 \leq i < k$, et $k$ vers 0), et soit $\rho$ une affectation rendant $A$ vrai à l'état $k$ uniquement. À l'état 0, $\langle a^* \rangle A$ est vrai, et pour tout $j < k$, $\neg \underbrace{\langle a \rangle \ldots \langle a \rangle}_{j \text{ times}} A$ est vrai aussi, puisque $A$ n'est pas vrai à l'état $j$. Donc il s'agit d'un modèle de $\Delta$, mais ce n'est pas un modèle de $\Phi$ (car $\Phi$ n'a pas de modèle), et donc $\Delta \not\models \Phi$. $\square$

Mais le $\mu$-calcul modal reste décidable :

**Théorème 11 (Decidabilité)** *Le problème de la satisfiabilité des formules du $\mu$-calcul modal est décidable.*

Le problème est en fait DEXPTIME-complet, mais c'est très difficile à montrer : les filtrations ne fonctionnent plus (bien que la propriété de petit modèle soit toujours vraie) et aucune axiomatisation finie complète n'est connue pour cette logique [KT90].

# 4 Vérification de modèles

De les logiques de la sorte que nous avons décrite en section 3, le problème de la satisfiabilité ou celui de la validité ne sont pas aussi importants que dans d'autres logiques. Une formule $\Phi$ dans ces logiques $L$ est une *spécification* d'une propriété souhaitée d'un programme. Les réalisations sont des systèmes de transitions (des programmes CCS ou des diagrammes états-transitions pour le matériel par exemple), autrement dit ce sont des *modèles* (des cadres et des affectations de Kripke). Vérifier la satisfiabilité d'une telle formule signifie chercher s'il est possible de réaliser cette spécification. Bien que ceci soit utile, ce n'est pas réaliste en général, parce que cela demande que le vérificateur de satisfiabilité invente un programme réalisant la spécification.

Le vrai problème auquel on est usuellement confronté est le suivant : étant donnée une spécification d'un système (sous forme du formule modale $\Phi$), et une réalisation de ce système (un modèle $(W, R, \rho)$), le programme satisfait-il la spécification ? En termes logiques :

$$(W, R, \rho) \text{ est-il un modèle de } \Phi \text{ ?}$$

Ce problème est appelé *vérification de modèle* ("model-checking" en anglais).

## 4.1 Machines séquentielles et vérification de modèle

L'exemple typique de vérification de modèle vient du domaine de la vérification de matériel. Les réalisations, ou circuits, sont décrits comme des machines séquentielles ayant $m$ bits d'entrée, $n$ bits de sortie, et $p$ bits d'état interne. Les machines séquentielles fonctionnent au rythme d'une horloge externe, et à chaque tic de l'horloge, la relation de transition $\delta$ de la machine est calculée, prenant en argument l'entrée courante et l'état interne courant, pour fournir la sortie et l'état interne suivants, comme illustré en figure 1. La relation de transition $\delta$ peut envoyer chaque combinaison entrée/état courant vers plusieurs combinaisons sorte/état suivant, ce qui signifie que la machine peut être non déterministe.

Formellement, la relation de transition est une relation $\delta$ de $\mathbb{B}^{m+p}$ vers $\mathbb{B}^{n+p}$. Une machine séquentielle définit un cadre de Kripke dont les mondes sont les $(m+n+p)$-uplets de booléens, ou bits, représentant l'entrée, la sortie et l'état courant, et dont la relation d'accessibilité $R$ est donnée par $(i_1, \ldots, i_m, s_1, \ldots, s_p, o_1, \ldots, o_n)\ R\ (i_1, \ldots, i_m, s'_1, \ldots, s'_p, o_1, \ldots, o_n)$ si et seulement si $(i_1, \ldots, i_m, s_1, \ldots, s_p)\ \delta\ (s'_1, \ldots, s'_p, o_1, \ldots, o_n)$.

Nous décrivons maintenant la technique de *vérification symbolique de modèle* ("symbolic model-checking"). Cette technique, utilisée avec des BDD, a rencontré un franc succès en vérification de matériel. Pour plus de détails, voir [McM93]. Nous donnons l'exemple de la vérification symbolique de modèle de formules du $\mu$-calcul modal avec une seule modalité (que nous écrirons $\Box$, comme en S4, mais signifiant "à tout état immédiatement successeur") par rapport aux machines séquentielles.

Remarquer que le cadre de Kripke est proprement gigantesque : il a $2^{m+n+p}$ états, et donc il est impossible de le représenter sous forme de graphe en mémoire même pour de petites valeurs de $m$, $n$, et $p$. Mais nous pouvons le représenter par une formule propositionnelle, que nous noterons encore $\delta$, construire à l'aide de $m+n+2p$ variables $i_1, \ldots, i_m, s_1, \ldots, s_p, s'_1, \ldots, s'_p, o_1, \ldots, o_n$. La taille de cette formule est grosso modo proportionnelle au nombre de portes dans le circuit codant la machine séquentielle, et est donc beaucoup plus acceptable.

Les variables $s_1, \ldots, s_p$ sont appelées *variables d'état*. Un état de la machine séquentielle est identifié à une affectation des variables d'état, d'entrée et de sortie. Toute formule propositionnelle, et même toute formule propositionnelle quantifiée sur l'ensemble des variables d'entrée, d'état et de sortie dénote un ensemble de configurations entrée/état/sortie (ou mondes) : l'ensemble des affectations qui satisfont la formule. Par exemple, si $A$ et $B$ sont deux variables d'état, alors $A \vee B$ représente l'ensemble des états dans lesquels $A$ est vraie ou bien $B$ est vraie.

L'idée fondamentale de la vérification symbolique de modèle est : étant donné un cadre de Kripke représenté comme une formule propositionnelle $\delta$ portant sur les variables d'entrée, d'état courant, d'état suivant et de sortie, et étant donnée une formule modale $\Phi$ à vérifier sur le cadre précédent, on calcule une formule propositionnelle (non modale) $\Psi$ portant sur les variables d'entrée, d'état et de sortie qui représente l'ensemble des mondes où la formule modale $\Phi$ est vraie. Alors, $\Phi$ est valide dans ce cadre de Kripke si et seulement si $\Psi$ est valides.

Nous devons récapituler et introduire quelques notions sur les formules propositionnelles (non modales). Si $\Phi$ est une formule propositionnelle, et $A$ est une variable propositionnelle, $\forall A \cdot \Phi$ est la formule $\Phi[\mathbf{T}/A] \wedge \Phi[\mathbf{F}/A]$, et $\exists A \cdot \Phi$ est la formule $\Phi[\mathbf{T}/A] \vee \Phi[\mathbf{F}/A]$. Observons aussi que :

**Lemme 12** *Soit $\Phi$ une formule propositionnelle, et $A$ a variable propositionnelle, telles que $A \Rightarrow A'$ implique $\Phi \Rightarrow \Phi[A'/A]$. (Nous disons que $\Phi$ est* monotone *en $A$.)*

*Alors il existe une formule propositionnelle $\mu A \cdot \Phi$ (resp. $\nu A \cdot \Phi$) telle que :*

- *$\mu A \cdot \Phi$ est logiquement équivalent à $\Phi[(\mu A \cdot \Phi)/A]$ (resp. $\nu A \cdot \Phi$ est logiquement équivalent à $\Phi[(\nu A \cdot \Phi)/A]$),*
- *et pour toute $\Phi'$ telle que $\Phi'$ est logiquement équivalente à $\Phi[\Phi'/A]$, alors $\mu A \cdot \Phi$ implique $\Phi'$ (resp. $\Phi'$ implique $\nu A \cdot \Phi$).*

**Preuve :** Soit $\mathcal{A}$ l'ensemble des affectations portant sur les variables libres dans $\Phi$.

Soit $f$ l'application de $\mathcal{P}(\mathcal{A})$ vers $\mathcal{P}(\mathcal{A})$ définie comme suit : si $S$ est l'ensemble des affectations satisfaisant une formule $\Phi'$, alors $f(S)$ est l'ensemble des affectations satisfaisant $\Phi[\Phi'/A]$. C'est bien une application (le domaine de $f$ est bien $\mathcal{P}(\mathcal{A})$), car tout $S$ peut être décrit comme l'ensemble des affectations satisfaisant une certaine formule $\Phi'$ construite sur $\mathcal{A}$: soit $\Phi'$ la disjonction de tous les $A_1^\rho \wedge A_2^\rho \wedge \ldots \wedge A_n^\rho$, où $\rho$ varie dans $S$, $\mathcal{A} = \{A_1, \ldots, A_n\}$ et $A^\rho$ vaut $A$ si $\rho(A)$ est vrai et $\neg A$ sinon.

Comme $\Phi$ est monotone par rapport à $A$, $f$ est une fonction croissante de $\mathcal{P}(\mathcal{A})$ vers $\mathcal{P}(\mathcal{A})$. Par le théorème du point fixe de Knaster et Tarski, $f$ a un plus petit point fixe $\underline{S}$ et un plus grand point fixe $\overline{S}$. (Ce théorème dit que l'ensemble des points fixes d'une application croissante d'un treillis complet, par exemple $\mathcal{P}(\mathcal{A})$, dans lui-même en est un sous-treillis non vide.) Soit $\mu A \cdot \Phi$ n'importe quelle formule satisfaite exactement par les affectations de $\underline{S}$, et $\nu A \cdot \Phi$ n'importe quelle formule satisfaite exactement par les affectations de $\overline{S}$. $\Box$

La preuve du lemme donne un moyen de calculer les plus petits et plus grands points fixes de fonctions propositionnelles monotones. Dans le cas de treillis finis (et $\mathcal{P}(\mathcal{A})$ est bien fini), la preuve du théorème de Knaster et Tarski est constructive. Le plus petit point fixe de $f$ est obtenu comme l'union de $f(\emptyset)$, $f(f(\emptyset))$, ..., $\underbrace{f(\ldots f(}_{k \text{ times}}\emptyset)\ldots)$, ..., cette suite devenant stationnaire au plus tard pour $k \geq 2^n$, où $n$ est le nombre de variables libres de $\Phi$. De même, le plus grand point fixe de $f$ est obtenu comme l'intersection de $f(\mathcal{A})$, $f(f(\mathcal{A}))$, ..., $\underbrace{f(\ldots f(}_{k \text{ times}}\mathcal{A})\ldots)$, ..., cette suite étant stationnaire au plus tard pour $k \geq 2^n$. Traduisons cette construction dans le langage des formules; nous pouvons donc calculer $\mu A \cdot \Phi$ comme suit :

1. Initialiser $\Phi'$ à $\mathbf{F}$;
2. calculer $\Phi'' = \Phi' \vee \Phi[\Phi'/A]$;
3. si $\Phi''$ est logiquement équivalent à $\Phi'$, alors retourner $\Phi'$;
4. sinon, soit $\Phi'$ égale $\Phi''$, et revenir à l'étape 2.

et de même, nous pouvons calculer $\nu A \cdot \Phi$ comme suit :

1. Initialiser $\Phi'$ à $\mathbf{T}$;
2. calculer $\Phi'' = \Phi' \wedge \Phi[\Phi'/A]$;
3. si $\Phi''$ est logiquement équivalente à $\Phi'$, alors retourner $\Phi'$;
4. sinon, soit $\Phi'$ égale $\Phi''$, et revenir à l'étape 2.

L'essentiel de la vérification symbolique de modèle est alors, comme McMillan en fait la remarque :

**Théorème 13** *Soit $(W, \delta)$ un cadre de Kripke, où les mondes sont des affectations portant sur les variables d'entrée $i_k$, $1 \leq k \leq m$, les variables d'état $s_k$, $1 \leq k \leq p$, et les variables de sortie $o_k$, $1 \leq k \leq n$; et où la relation de transition est codée comme une formule propositionnelle $\delta$ sur $i_k$, $s_k$, $o_k$ et les variables d'état suivant $s'_k$, $1 \leq k \leq p$.*

*Soit $\overline{\Phi}$ l'ensemble des affectations satisfaisant une formule propositionnelle quantifiée $\Phi$. Pour toute formule propositionnelle $\Phi$ portant sur $i_k$, $s_k$ et $o_k$ uniquement, et pour tout monde $w$, $w \models \Phi$ est caractérisé comme suit :*

- *$w \models A$ si et seulement si $w \in \overline{A}$, pour $A$ une des variables $i_k$, $s_k$ ou $o_k$;*
- *$w \models \Phi \vee \Phi'$ si et seulement si $w \in \overline{\Phi \vee \Phi'}$ (et de même pour $\wedge$, $\neg$, $\Rightarrow$);*
- *$w \models \Box\Phi$ si et seulement si $w \in \overline{\forall s'_1 \cdot \ldots \forall s'_p \cdot \delta \Rightarrow \Phi[s'_1/s_1, \ldots, s'_p/s_p]}$;*
- *$w \models \mu A \cdot \Phi$ si et seulement si $w \in \overline{\mu A \cdot \Phi}$;*
- *$w \models \nu A \cdot \Phi$ si et seulement si $w \in \overline{\nu A \cdot \Phi}$;*

*où dans les deux derniers cas, nous avons supposé que $\Phi$ était monotone en $A$.*

Le théorème fournit la traduction souhaitée des formules modales $\Phi$ vers les formules propositionnelles non modales $\Psi$ telle que les configurations de $(W, \delta)$ où $\Phi$ est vraie sont représentées exactement par les affectations satisfaisant $\Psi$. L'algorithme de traduction fonctionne comme suit :

- si $\Phi$ est une variable $A$, retourner $A$;
- si $\Phi = \Phi' \vee \Phi''$, soit $\Psi'$ la traduction de $\Phi'$ et $\Psi''$ la traduction de $\Phi''$, alors retourner $\Psi' \vee \Psi''$; (de même pour $\wedge$, $\neg$, $\Rightarrow$;)
- si $\Phi = \Box\Phi'$, soit $\Psi'$ la traduction de $\Phi'$, et retourner $\forall s'_1 \cdot \ldots \forall s'_p \cdot \delta \Rightarrow \Psi'[s'_1/s_1, \ldots, s'_p/s_p]$;
- si $\Phi = \mu A \cdot \Phi'$, soit $\Psi'$ la traduction de $\Phi'$, et retourner $\mu A \cdot \Psi'$;
- si $\Phi = \nu A \cdot \Phi'$, soit $\Psi'$ la traduction de $\Phi'$, et retourner $\nu A \cdot \Psi'$.

Que l'algorithme est correct est facile à montrer, sauf deux points liés aux formules en $\mu$ et $\nu$. D'abord, la variable $A$ n'est une variable ni d'entrée, ni d'état, ni de sortie, ce qui, à strictement parler, nous empêche d'utiliser le théorème 13. Nous corrigeons ce problème en ajoutant $A$ à l'ensemble des variables d'entrée par exemple : on vérifiera que $\delta$ ne change pas. Ensuite, il n'est pas clair que les traductions des opérateurs $\mu$ et $\nu$ soient bien définies, parce que nous devons d'abord nous assurer que la formule $\Psi'$ est monotone en $A$. La raison en est le lemme facile qui suit :

**Lemme 14** *Soit $\Phi'$ une formule modale, où $A$ n'apparaît que positivement. Soit $\Psi'$ la traduction de $\Phi'$. Alors $\Psi'$ est monotone en $A$.*

## 4.2 Diagrammes de décision binaire

Pour savoir si la formule modale $\Phi$ est valide dans le cadre de Kripke donné, il est nécessaire et suffisant de vérifier que la formule traduite $\Psi$ est valide.

Cependant, le calcul des points fixes $\mu A \cdot \Psi$ et $\nu A \cdot \Psi$ est ardu, parce que nous devons faire un test d'équivalence logique à chaque passage de la boucle. Remarquer que la vérifier de l'équivalence de deux formules $\Psi_1$ et $\Psi_2$ revient à vérifier la validité de $\Psi_1 \Leftrightarrow \Psi_2$, et ceci peut prendre un temps exponentiel en les tailles de $\Psi_1$ et $\Psi_2$.

Nous pouvons économiser une grande partie de ce coût dans les applications pratiques si nous représentons les formules propositionnelles sous une forme canonique sur laquelle les opérations propositionnelles sont efficaces. Les diagrammes de décision binaire (BDD) ont toutes les propriétés souhaitées. En particulier, la négation s'effectue en temps linéaire (même contant si on utilise des TDG, un raffinement des BDD), et toutes les opérations binaires opèrent en temps proportionnel à $smn$, où $m$ et $n$ sont les tailles des BDD à combiner, et $s$ est le coût du partage.

Pour utiliser les BDD efficacement, cependant, il nous faut encore définir la notion de substitution sur les BDD. Sur les formules construites à l'aide de $\vee$, $\wedge$, $\neg$, $\Rightarrow$, la substitution était juste le remplacement textuel. Mais si $\Phi$ est un BDD, nous voulons que $\Phi[\Phi'/A]$ soit le BDD de la formule $\Psi[\Phi'/A]$, où $\Psi$ est n'importe quelle formule représentée par le BDD $\Phi$. Ceci n'est pas un simple remplacement textuel, qui ne fournit pas un BDD en général.

Nous définissons d'abord $\Phi[\mathbf{T}/A]$ et $\Phi[\mathbf{F}/A]$ par récurrence structurelle sur $\Phi$:

- $\mathbf{T}[\mathbf{T}/A] = \mathbf{T}$, $\mathbf{F}[\mathbf{T}/A] = \mathbf{F}$, $\mathbf{T}[\mathbf{F}/A] = \mathbf{T}$, $\mathbf{F}[\mathbf{F}/A] = \mathbf{F}$;
- si $\Phi = A \longrightarrow \Phi_+; \Phi_-$, alors $\Phi[\mathbf{T}/A] = \Phi_+$, $\Phi[\mathbf{F}/A] = \Phi_-$;
- si $\Phi$ est de la forme $A' \longrightarrow \Phi_+ : \Phi_-$, avec $A' \neq A$,
  alors $\Phi[\mathbf{T}/A] = BDDmake(A', \Phi_+[\mathbf{T}/A], \Phi_-[\mathbf{T}/A])$, et $\Phi[\mathbf{F}/A] = BDDmake(A', \Phi_+[\mathbf{F}/A], \Phi_-[\mathbf{F}/A])$.

Nous pouvons alors calculer $\Phi[\Phi'/A]$ comme $(\Phi' \Rightarrow \Phi[\mathbf{T}/A]) \wedge (\neg\Phi' \Rightarrow \Phi[\mathbf{F}/A])$, ou comme $(\Phi' \wedge \Phi[\mathbf{T}/A]) \vee (\neg\Phi' \wedge \Phi[\mathbf{F}/A])$.

C'est tout ce dont nous avons besoin pour calculer des quantifications propositionnelles et des points fixes : nous avons tous les outils nécessaires à la vérification symbolique de modèle à l'aide de BDD.

### 4.3 Développements

Le codage de la relation de transition $\delta$ comme une formule propositionnelle entre $m$ variables d'entrée, $2p$ variables d'état courant et suivant, et $n$ variables de sortie n'est pas économique. En particuler, si le circuit est *déterministe*, $\delta$ est en fait une fonction de $\mathbb{B}^{m+p}$ (entrée et état courant) vers $\mathbb{B}^{p+n}$ (état suivant et sortie). Alors on peut représenter $\delta$ comme un $(p+n)$-uplet de fonctions $\delta_1, \ldots, \delta_{p+n}$ de $\mathbb{B}^{m+p}$ vers $\mathbb{B}$, autrement dit comme un $(p+n)$-uplet de formules sur $m+p$ variables. Ceci est très important, car la taille d'un BDD sur $k$ variables peut atteindre une taille de l'ordre de $2^k$. (Plus précisément, la borne supérieure est $2^k/k$.) Par exemple, dans le cas $m = n = 0$, le BDD pour $\delta$ sur les $2p$ variables d'état courant et suivant peut avoir une taille proportionnelle au carré de la plus grande taille des BDD de $\delta_i$, $1 \leq i \leq p+n$ : en pratique, la représentation sous forme de $(p+n)$-uplets permet de vérifier des circuits en quelques mégaoctets de mémoire qui nécessiteraient plus gigaoctets ou téraoctets dans la représentation précédente. (Ceci sans parler du temps qu'il nous faudrait pour construire de telles structures en mémoire.)

Si le circuit est non déterministe, nous pouvons encore utiliser une variante de la représentation sous forme de $(p+n)$-uplet. Nous pouvons en général décrire le degré de non-déterminisme en quelques bits seulement : c'est-à-dire, nous savons qu'il n'y aura jamais plus de, disons, $2^k$ mondes en relation avec un monde donné quelconque, où $k$ est usuellement petit (au moins comparé à $p$). Nous créons alors $k$ variables propositionnelles auxiliaires $x_1, \ldots, x_k$, et codons $\delta$ comme l'union de $2^k$ fonctions paramétrées $\delta_\rho$, où $\rho$ est une affectation de valeurs de vérité à $x_i$, $1 \leq i \leq k$.

La vérification de modèle se pratique comme avant, sauf que le troisième cas du théorème 13 doit devenir :

- si $\Phi = \Box\Phi'$, soit $\Psi'$ la traduction de $\Phi'$,
  et retourner $\forall x_1 \cdot \ldots \forall x_k \cdot \Psi'[\delta_1(i_1, \ldots, i_m, s_1, \ldots, s_p)/s_1, \ldots, \delta_p(i_1, \ldots, i_m, s_1, \ldots, s_p)/s_p]$;

Remarquer que dans le cas déterministe, nous n'avons même plus besoin de quantification.

Une autre variante de la méthode de base porte sur la façon dont nous représentons les ensembles de configurations. Nous avons utilisé des formules $\Phi$ représentant des ensembles de configurations comme l'ensembles des affectations qui les satisfont. C'est-à-dire, si l'on considère $\Phi$ comme une application de $\mathbb{B}^{m+p+n}$ vers $\mathbb{B}$, que nous représentons des ensembles de configurations comme l'image inverse $\Phi^{-1}(\{\mathbf{T}\})$. Une façon de faire duale est de prendre un $(m+p+n)$-uplet d'applications booléennes $\Phi_i$, $1 \leq i \leq m+p+n$, de $\mathbb{B}^{m+p}$ vers $\mathbb{B}$, et de représenter des ensembles de configurations comme l'image directe de $\mathbb{B}^{m+p}$ par ce $(m+p+n)$-uplet

Finalement, nous devons mentionner que vérifier qu'une formule modale $\Phi$ est toujours vraie (dans tous les mondes) n'est pas en général ce que nous voulons. Une machine séquentielle démarre en effet dans une configuration donnée (disons, toutes les entrées et tous les bits d'état mis à $\mathbf{F}$), et alors elle ne passe que par les configurations *atteignables* depuis cette configuration initiale. Par exemple, un compteur à 2 bits censé compter 0, 1, 2, 0, 1, 2, etc. (correspondant aux configurations $(\mathbf{F}, \mathbf{F})$, $(\mathbf{F}, \mathbf{T})$, $(\mathbf{T}, \mathbf{F})$ respectivement) n'atteindra jamais la configuration où les deux bits sont $\mathbf{T}$. Ce que nous voulons en fait, c'est vérifier que la formule modale $\Phi$ est vraie dans tous les états atteignables, pas nécessairement dans tous les mondes.

Une façon naïve de le faire est de calculer la traduction non modale $\Psi$ de $\Phi$ par rapport au cadre de Kripke donné comme avant, ensuite de calculer une formule $\Psi'$ telle que les affectations qui la satisfont sont exactement les configurations atteignables, et alors de vérifier que $\Psi' \Rightarrow \Psi$ est valide. Ceci est cependant trop coûteux en pratique, et il est intéressant d'élaguer le BDD $\Psi$ en utilisant l'ensemble des configurations atteignables lors de sa construction.

On trouvera des discussions plus poussées de la vérification de modèle à l'aide de BDD et des applications dans [McM93]. [CM95] est une bonne source de références bibliographiques. Un traitement plus en profondeur des logiques modales et temporelles en informatique se trouve dans [Sti92, Eme90]. Les logiques modales sont le sujet de [GG84].

## References


[Che80] Brian F. Chellas. *Modal Logics: an Introduction.* Cambridge University Press, 1980.

[CM95] Olivier Coudert and Jean-Christophe Madre. The implicit set paradigm: A new approach to finite state system verification. *Formal Methods in System Design*, 6(2):133–145, 1995.

[Eme90] E. Allen Emerson. Temporal and modal logics. In Leeuwen [Lee90], chapter 16, pages 995–1072.

[Fit83] Melvin C. Fitting. *Proof Methods for Modal and Intuitionistic Logics.* Reidel Publishing Company, Dordrecht, 1983.

[GG84] Dov M. Gabbay and F. Guenthner. *Handbook of Philosophical Logic*, volume 2. Reidel Publishing Company, Dordrecht, 1984.

[Gor91] Rajeev Prabhakar Goré. Semi-analytic tableaux for propositional normal modal logics with applications to non-monotonicity. *Logique et Analyse*, 133–134:73–104, 1991.

[HC68] G.E. Hughes and M.J. Creswell. *An Introduction to Modal Logics.* Methuen and Co., 1968.

[HC84] G.E. Hughes and M.J. Creswell. *A Companion to Modal Logics.* Methuen and Co., 1984.

[KT90] Dexter Kozen and Jerzy Tiuryn. Logics of programs. In Leeuwen [Lee90], chapter 14, pages 789–840.

[Lee90] Jan van Leeuwen, editor. *Handbook of Theoretical Computer Science*, volume B. Elsevier Science Publishers b.v., 1990.

[Lew18] Clarence I. Lewis. *A Survey of Symbolic Logic.* University of California, 1918.

[McM93] Kenneth L. McMillan. *Symbolic Model Checking.* Kluwer Academic Publishers, 1993.

[Mil89] Robin Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice Hall, 1989.

[Ohl93] Hans-Jürgen Ohlbach. Translation methods for non-classical logics: An overview. *Bulletin of the Interest Group in Pure and Applied Logics*, 1(1):69–89, 1993. Available by ftp on the IGPL server, at `theory.doc.ic.ac.uk:/theory/forum/igpl/Bulletin`.

[Sti92] Colin Stirling. Modal and temporal logics. In Samson Abramsky, Dov M. Gabbay, and Thomas S. E. Maibaum, editors, *Handbook of Logics in Computer Science*, volume 2, pages 477–563. Clarendon Press, Oxford, 1992.