Logiques modales et méthode des tableaux: une introduction

Stéphane Demri

demri@lsv.ens-cachan.fr

LSV, ENS de Cachan

MPRI, janvier 2005

Plan du cours

- Une brève introduction aux logiques modales
  - Syntaxe, sémantique et problèmes
  - Exemples
  - Calculs à la Hilbert
  - Traduction vers la logique classique

- Algorithmes de Ladner
  - Pour la logique K
  - Borne **PSPACE** de complexité
  - Pour la logique S4 avec détection de cycles

- Automates d'arbres.

Plan du cours (suite)

- Calculs de F. Massacci avec préfixes
  - Calculs pour les logiques K,B,S4 et S5
  - Complétude (idées)
  - Procédures de décision
- Calculs de séquents
  - pour les logiques K et S4
  - relation avec les tableaux

Des modalités

- Modalité: terme modifiant le rapport entre un prédicat et un sujet
  - addition d'adverbe: "Alice viendra certainement"
  - proposition complétive: "Il est certain que Jean viendra"
- Exemples:
  - modalité ontique: "Il est nécessaire que ..." ($\Box \ldots$)
  - modalité temporelle: "Demain, ..." ($\mathsf{X} \ldots$)
  - modalité épistémique: "Alice croit que Paul sait que ..." ($B_{Alice}\ K_{Paul}\ \ldots$)
  - modalité déontique: "Il est obligatoire ..." ($O \ldots$), "Il est permis ..." ($P \ldots$).
- Traditionnellement, la logique modale est la logique de la nécessité et de la possibilité mais il s'agit d'une vue dépassée en informatique.

Autres modalités en informatique

- $p \Rightarrow [\pi]q$: “Si p est vrai avant l’exécution du programme π, alors q est vrai après”
- $\forall$F p: “Pour toutes les exécutions, il existe un état vérifiant p”
- $C_{A,B}\ p \Rightarrow K_A\ q$: “Si p fait partie de la connaissance commune des agents A et B alors l’agent A sait que q est vrai”
- $[p]\langle q^{-1}\rangle racine$:”Toute position atteinte à partir de la racine d’un document en empruntant un chemin étiqueté par un mot de p peut être atteinte par un chemin étiqueté par un mot de q”

$$\begin{tikzcd} \text{racine} \arrow[r, bend left, "p"] \arrow[r, bend right, "q"'] & y \end{tikzcd}$$

Langages modaux

- Simples et suffisament expressifs pour parler des structures relationnelles de la forme $\langle W, R_1, R_2, \ldots \rangle$.

- Vision locale de la description des structures.

- Domaines d'application
  - raisonner sur la connaissance, obligation, croyance
  - informatique: logiques temporelles, dynamiques . . .
  - mathématique: raisonnement géométrique, théorie des ensembles, arithmétique
  - linguistique: sémantique du langage naturel

Qu'est-ce qu'une logique modale?

- Approches standard:
  - axiomatique: calculs à la Frege - Hilbert
  - sémantique: mondes possibles, systèmes de transitions étiquetées, . . .
  - algébrique: algèbres modales comme extensions des algèbres de Boole
- Aujourd'hui, les logiques modales sont considérées comme
  - des logiques munies d'opérateurs modaux (temporels, épistémiques, terminologiques, dynamiques, . . .)
  - des langages pour décrire des structures relationnelles
  - des fragments pertinents de la logique classique (1er ou 2nd ordre)

Langage modal et modèles

- Langage propositionnel de base:

$$\phi \ ::= \ \overbrace{p \ | \ \neg\phi \ | \ \phi \wedge \phi'}^{\text{calcul propositionnel}} \ | \ \overbrace{\Box\phi \ | \ \Diamond\phi}^{\text{partie modale}}$$

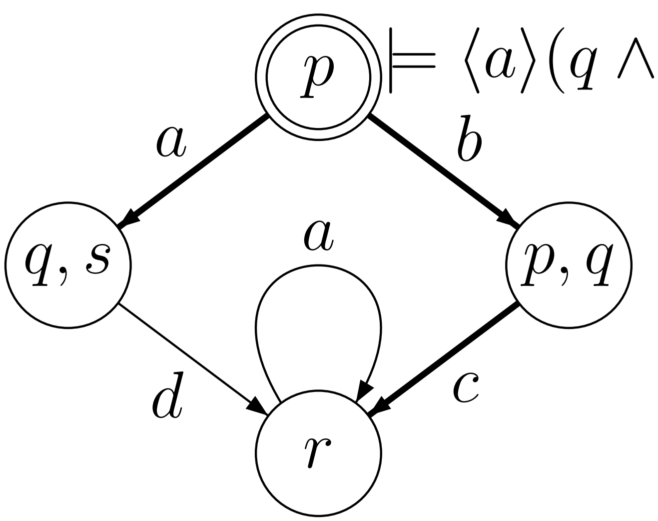
  $p \in \mathrm{PROP}$ et $\bot, \top, \vee, \Rightarrow$ sont des abréviations.

- Modèle de Kripke: $\mathcal{M} = \langle W, R, v \rangle$:
  – W: ensemble non-vide
  – R: relation binaire sur W
  – $v : \mathrm{PROP} \to \mathcal{P}(W)$: fonction d'interprétation

- Graphe orienté dont chaque noeud est une interprétation du calcul propositionnel.

Relation de satisfaction $\models$

- $\mathcal{M}, w \models p_i \overset{\text{def}}{\Leftrightarrow} w \in v(p_i)$; $\mathcal{M}, w \models \neg\phi \overset{\text{def}}{\Leftrightarrow} \mathcal{M}, w \not\models \phi$;
- $\mathcal{M}, w \models \phi \wedge \phi' \overset{\text{def}}{\Leftrightarrow} \mathcal{M}, w \models \phi$ et $\mathcal{M}, w \models \phi'$;
- $\mathcal{M}, w \models \Box\phi \overset{\text{def}}{\Leftrightarrow}$ pour tous les $w' \in R(w)$, $\mathcal{M}, w' \models \phi$;
- $\mathcal{M}, w \models \Diamond\phi \overset{\text{def}}{\Leftrightarrow}$ il existe $w' \in R(w)$ tel que $\mathcal{M}, w' \models \phi$.



$\models \langle a\rangle(q \wedge [d]r) \wedge [b][c][a]\neg q$

Problèmes

- $\mathcal{C}$: classe de modèles définissant une logique modale.

- satisfaisabilité: ϕ est $\mathcal{C}$*-satisfaisable* $\overset{\text{def}}{\Leftrightarrow}$ il existe $\mathcal{M} \in \mathcal{C}$ et $w \in \mathcal{M}$ tels que $\mathcal{M}, w \models \phi$.

- validité: ϕ est $\mathcal{C}$*-valide* $\overset{\text{def}}{\Leftrightarrow}$ $\neg\phi$ n'est pas $\mathcal{C}$-satisfaisable.

- model-cheking: $\mathcal{M}, w \models \phi$? avec $\mathcal{M}$ fini.

- Une logique modale peut-être définie comme un langage et une classe $\mathcal{C}$ de modèles relative au langage.

Quelques logiques modales standard

- La logique K est caractérisée par la classe de tous les modèles. (satisfaisabilité **PSPACE**-complète).
- La logique T est caractérisée par la classe de tous les modèles réflexifs.
- La logique S4 est caractérisée par la classe de tous les modèles réflexifs et transitifs.
- La logique S5 est caractérisée par la classe de tous les modèles dont la relation R est une relation d'équivalence. (satisfaisabilité **NP**-complète).
- La logique G est caractérisée par la classe de tous les modèles dont la relation R est transitive et sans chemin infini.
- En fait, il existe de nombreux familles de logiques modales (logiques dynamiques, logiques terminologiques, logiques temporelles, ...) et de nombreux hiérarchies infinies de logiques modales.

Calcul de Hilbert pour K

Le système modal à la Hilbert K (en l'honneur de S. Kripke) est composé des schémas d'axiomes

1. les tautologies du calcul propositionnel;
2. $\Box p \Rightarrow (\Box(p \Rightarrow q) \Rightarrow \Box q)$;
3. $\Diamond p \Leftrightarrow \neg\Box\neg p$;

et des règles d'inférence:

1. *modus ponens*:

$$\frac{p \quad p \Rightarrow q}{q}$$

2. *nécessitation*:

$$\frac{p}{\Box p}$$

Dérivations

- Une *dérivation* du système K est une séquence $\langle \phi_1, \ldots, \phi_n \rangle$ telle pour $i \in \{1, \ldots, n\}$,
  - soit ϕ_i est une instance d'un schéma d'axiomes de K;
  - soit il existe $j < i$ tel que $\phi_i = \Box \phi_j$;
  - soit il existe $j, j' < i$ tels qu'appliquer le *modus ponens* sur ϕ_j and $\phi_{j'}$ produit la formule ϕ_i.
- Les théorèmes de K sont les formules apparaissant dans une dérivation de K.
- A la place des schémas d'axiomes (1) on peut se restreindre aux trois schémas d'axiomes suivants sans affaiblir le pouvoir déductif du système:
  1. $p \Rightarrow (q \Rightarrow p)$;
  2. $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$;
  3. $(\neg p \Rightarrow \neg q) \Rightarrow (q \Rightarrow p)$.

Admissibilité

Lemme. La règle

$$\frac{\phi \Rightarrow \psi}{\Box\phi \Rightarrow \Box\psi}$$

est admissible (ajouter cette règle n'augmente pas le pouvoir déductif du système).

- Supposons que $\vdash_K \phi \Rightarrow \psi$.
- En appliquant la règle de nécessitation, nous obtenons $\vdash_K \Box(\phi \Rightarrow \psi)$.
- $(p \Rightarrow (q \Rightarrow q')) \Rightarrow (q \Rightarrow (p \Rightarrow q'))$ est une tautologie.
- Remplacer p [resp. q, q'] par $\Box\phi$ [resp. $\Box(\phi \Rightarrow \psi)$, $\Box\psi$]:

$$\vdash_K (\Box\phi \Rightarrow (\Box(\phi \Rightarrow \psi) \Rightarrow \Box\psi)) \Rightarrow (\Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi))$$

Admissibilité (suite)

- Remplacer p [resp. q, q'] par $\Box\phi$ [resp. $\Box(\phi \Rightarrow \psi)$, $\Box\psi$]:

$$\vdash_K (\Box\phi \Rightarrow (\Box(\phi \Rightarrow \psi) \Rightarrow \Box\psi)) \Rightarrow (\Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi))$$

- Or, $\vdash_K \Box\phi \Rightarrow (\Box(\phi \Rightarrow \psi) \Rightarrow \Box\psi)$ en instanciant le schéma d'axiomes (2).

- On utilise alors la règle de *modus ponens* pour établir $\vdash_K \Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi)$.

- En utilisant à nouveau la régle de *modus ponens* sur $\Box(\phi \Rightarrow \psi)$ et $\Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi)$ on a donc $\vdash_K \Box\phi \Rightarrow \Box\psi$.

Complétude

Théorème. (Correction et complétude) $\vdash_K \phi$ ssi ϕ est valide dans tous les modèles.

- La preuve de correction est facile (par induction sur la longueur des dérivations).
- La preuve de complétude standard utilise le modèle canonique $\mathcal{M} = \langle W, R, v \rangle$:
  - W est l'ensemble des ensembles maximallement consistents par rapport à $\vdash_K$.
  - $XRY \overset{\text{def}}{\Leftrightarrow}$ si $\Box\phi \in X$ alors $\phi \in Y$ pour toute formule $\Box\phi$.
  - $v(p) \overset{\text{def}}{=} \{X \in W : p \in X\}$.
  - Propriété fondamentale: $\mathcal{M}, X \models \phi$ ssi $\phi \in X$.
- Ce calcul n'est pas adapté pour la mécanisation.

Extensions structurelles

- Famille de relations:

$$\langle W, (R_i)_{i \in I}, v \rangle, \quad ([i])_{i \in I}.$$

- Structurer le domaine:

$$\langle W_1 \times W_2, R_1, R_2, v \rangle, \quad R_i \subseteq W_i \times W_i$$

$\mathcal{M}, \langle w_1, w_2 \rangle \models [1]\phi \stackrel{\text{def}}{\Leftrightarrow}$ pour tous les $w_1' \in R_1(w_1)$,
$\mathcal{M}, \langle w_1', w_2 \rangle \models \phi$.

- Relations ternaires:
$\mathcal{M}, w \models \Diamond(\phi_1, \phi_2) \stackrel{\text{def}}{\Leftrightarrow}$ il existe w', w'' tels que $R(w, w', w'')$,
$\mathcal{M}, w' \models \phi_1$ et $\mathcal{M}, w'' \models \phi_2$.

Contraintes sur les modèles

Plutôt utilisées pour les problèmes de satisfaisabilité

- Contraintes relationnelles:
  - PDL: $R_{\pi^*} = (R_\pi)^* \overset{\text{def}}{=} \bigcup\{R_\pi^i : i \geq 0\}$
  - "tense logic" de A. Prior avec passé P et futur F
  - opérateur modal universel $[U]$: $\mathcal{M}, w \models [U]\phi \overset{\text{def}}{\Leftrightarrow}$ pour tous les $w' \in W$, on a $\mathcal{M}, w' \models \phi$.
    On perd ici le caractère local des modalités.
- Contraintes propositionnelles:
  - nominaux: i est un nominal si on impose que $v(i)$ soit un singleton
  - logique intuitionniste: les propositions atomiques sont persistentes ($w \in v(p)$ et $\langle w, w' \rangle \in R$ impliquent $w' \in v(p)$)

Autres opérateurs

Plutôt utilisées dans les logiques temporelles pour la vérification formelle de programmes.

- Quantifications sur les chemins, le long des chemins CTL, CTL*, etc $\forall\phi$Uψ.
- Opérateurs de point fixe: μ-calcul.
- Dans l'absolu on peut regagner le pouvoir d'expression de la logique du premier ordre ou du second ordre.

Contraintes de chemins

- Graphe étiqueté avec racine: $G = \langle W, rac, (R_a)_{a \in L} \rangle$ avec L un ensemble d'étiquettes de transitions.
- Expressions de chemins:

$$a \in L \;\mid\; \sharp \;\mid\; p_1 ; p_2 \;\mid\; p_1 \cup p_2 \;\mid\; p^*$$

- Interprétation de p dans G:

$$\begin{array}{rcl}
tr(a) & = & R_a \text{ pour } a \in L \\
tr(p^*) & \text{est} & \text{la fermeture réflexive et transitive de } tr(p) \\
tr(\#) & = & \bigcup_{a \in L} R_a \\
tr(\epsilon) & = & \{\langle u, u \rangle : u \in V\} \\
tr(p_1 \,;\, p_2) & = & \{\langle u, v \rangle : \exists z \, (tr(p_1)(u, z) \wedge tr(p_2)(z, v))\} \\
tr(p_1 + p_2) & = & tr(p_1) \cup tr(p_2)
\end{array}$$

Problème de l'implication

- Contrainte d'inclusion de chemins: $c = p \subseteq q$.

- $G \models p \subseteq q \overset{\text{def}}{\Leftrightarrow} tr(p)(rac) \subseteq tr(q)(rac)$.

- Problème de l'implication:

  entrée: contraintes d'inclusion $c_1, \cdots, c_{n+1}$, $n \geq 0$.

  question: est-ce que pour toute structure $G = \langle W, rac, (R_a)_{a\in} \rangle$,
  si $G \models c_1$ et $\cdots$ et $G \models c_n$ alors $G \models c_{n+1}$?
  (noté $c_1, \cdots, c_n \rightarrow c_{n+1}$)

- De nombreux fragments de ce problème sont dans **PSPACE**. Le problème est dans **EXPTIME** et on ignore aujourd'hui si le problème dans sa généralité est dans **PSPACE**.

Une variante de PDL

- Conception d'une logique PDL$^{\mathrm{path}}$ pour exprimer le problème de l'implication et bien d'autres problèmes.
- Expressions de chemins: comme précédemment + opérateur unaire $^{-1}$.
- Formule de PDL$^{\mathrm{path}}$:

$$\phi ::= \top \mid \perp \mid \mathtt{root} \mid \neg\phi \mid \phi \wedge \phi \mid \langle p\rangle\phi \mid [p]\phi.$$

- $G, w \models \mathtt{root} \overset{\text{def}}{\Leftrightarrow} w = rac$.
- $G, w \models \langle p\rangle\phi \overset{\text{def}}{\Leftrightarrow}$ il existe $w' \in tr(p)(w)$ tel que $G, w' \models \phi$ avec $tr(p^{-1}) = tr(p)^{-1}$.

Codage

- Les propriétés suivantes sont équivalentes:
  - $c_1, \cdots, c_n \longrightarrow c_{n+1}$
  - $[p_1]\langle q_1^{-1}\rangle\texttt{root} \wedge \cdots \wedge [p_n]\langle q_n^{-1}\rangle\texttt{root} \Rightarrow [p_{n+1}]\langle q_{n+1}^{-1}\rangle\texttt{root}$ est PDL$^{\mathrm{path}}$ valide avec $c_i = p_i \subseteq q_i$.

- Quelques problèmes ouverts:
  - Caratériser la complexité du problème de l'implication.
  - Définir une logique dans **PSPACE** qui code les fragments dans **PSPACE** du problème de l'implication.
  - Complexité de la version déterministe de PDL$^{\mathrm{path}}$.

Traduction relationnelle

- La sémantique des opérateurs modaux peut être codée dans la logique classique.

- Traduction:
  - $\mathrm{ST}(p_j, x_i) \overset{\text{def}}{=} P_j(x_i)$
  - $\mathrm{ST}(\neg\phi, x_i) \overset{\text{def}}{=} \neg\mathrm{ST}(\phi, x_i)$ (clause similaire pour $\wedge$)
  - $\mathrm{ST}(\Box\phi, x_i) \overset{\text{def}}{=} \forall\, x_{i+1}\ (R(x_i, x_{i+1}) \Rightarrow \mathrm{ST}(\phi, x_{i+1}))$
  - $\mathrm{ST}(\Diamond\phi, x_i) \overset{\text{def}}{=} \exists\, x_{i+1}\ R(x_i, x_{i+1}) \wedge \mathrm{ST}(\phi, x_{i+1})$

- Exemple:

$$\mathrm{ST}(\Diamond\Diamond p_1, x_0) \overset{\text{def}}{=} \exists x_1\ (R(x_0, x_1) \wedge (\exists x_2\ R(x_1, x_2) \wedge P_1(x_2)))$$

Préservation

- $\mathrm{ST}(\phi, x_i)$ a pour seule variable libre x_i.
- $\mathrm{ST}(\phi, x_0)$ se calcule en temps polynomial en $|\phi|$.
- $\mathcal{M}, w \models \phi$ ssi $\mathcal{M} \models_{fol} \mathrm{ST}(\phi, x_0)[x_0 \leftarrow w]$
- Si la classe $\mathcal{C}$ de modèles est définissable dans la logique classique avec la formule $\phi_{\mathcal{C}}$ alors

  ϕ est $\mathcal{C}$-satisfaisable ssi $\phi_{\mathcal{C}} \wedge\ \mathrm{ST}(\phi, x_0)$ est satisfaisable dans la logique classique.
- $\phi_{\mathrm{S4}} = \forall\ x\ R(x, x)\ \wedge\ \forall\ x, y, z\ R(x, y) \wedge R(y, z) \Rightarrow R(x, z)$.

Recyclage des variables dans FO2

- FO2 : logique classique avec égalité, constantes et au plus 2 variables
- Complexité: Problème de satisfaisabilité **NEXPTIME**-complet (FO3 est indécidable)
- Recyclage:
  - $\mathrm{ST}(\Box\phi, x) \stackrel{\text{def}}{=} \forall\, y\ (R(x,y) \Rightarrow \mathrm{ST}(\phi, y))$
  - $\mathrm{ST}(\Box\phi, y) \stackrel{\text{def}}{=} \forall\, x\ (R(y,x) \Rightarrow \mathrm{ST}(\phi, x))$
- Model-checking pour la logique classique
  - **PSPACE**-complet dans le cas général
  - en temps polynomial pour les fragments avec au plus k variables

Fragment gardé

- "Fragment modal" de la logique des prédicats.
- Fermeture par opérateurs Booléens et

$$\exists\overline{x}(\alpha(\overline{x},\overline{y}) \wedge \phi) \quad \forall\overline{x}(\alpha(\overline{x},\overline{y}) \Rightarrow \phi)$$

  - $\overline{x}$ et $\overline{y}$ sont des séquences de variables individuelles
  - $\alpha(\overline{x},\overline{y})$ est une formule atomique
  - chaque variable libre de ϕ est une variable de $\overline{x},\overline{y}$.

- $\mathrm{ST}(\Diamond\Diamond p_1, x_0) \stackrel{\text{def}}{=} \exists x_1\ (R(x_0,x_1) \wedge (\exists x_2\ R(x_1,x_2) \wedge P_1(x_2)))$ est dans GF.

Fragment gardé (suite)

- $\mathrm{ST}(\phi, x)$ appartient toujours à GF.
- Problème de satisfaisabilité restreint aux symboles de prédicat d'arité au plus $k \geq 2$, **EXPTIME**-complet.
- **2EXPTIME**-complétude du fragment gardé entier.
- GF avec relations transitives est indécidable.

Mécanisation et traduction

- Une logique peut-être caractérisée par une classe de modèles qui n'est pas définissable dans la logique du premier ordre:
  - fermeture transitive de relation (comme pour PDL)
  - relation sans chemin infini (comme pour G)
- Le fragment cible de la logique classique n'est pas toujours décidable.
  - GF avec relations transitives est indécidable.
  - FO3 est indécidable.
- Perte de structure de la formule modale initiale.

Réduction et traduction

Existence: le model-checking pour PLTL est dans **PSPACE** et la satisfaisabilité pour GF est **PSPACE**-difficile
$\rightarrow$ il existe un codage du model-checking pour PLTL vers un fragment décidable de la logique classique . . . via les machines de Turing !!

Expressivité: CTL* est plus expressive que PLTL mais . . .
il existe une réduction en temps polynomial du model-checking de CTL* vers le model-checking pour PLTL.

Second-ordre: la logique modale G est caractérisée par une classe de modèles du second ordre (R^{-1} bien fondée, transitive) mais . . .
il existe une réduction de la satisfaisabilité pour G vers la satisfaisabilité pour n'importe quel fragment de la logique classique **PSPACE**-difficile (GF, FO2, . . .).

Un fragment WLGF2 dans PSPACE

- Vocabulaire: deux variables individuelles x_0, x_1, des symboles de prédicats unaires et binaires.

- Fragment fermé par les opérateurs Booléens.

- Si $\phi(x_i)$ et $\psi(x_i, x_{1-i})$ sont des formules de WLGF2 pour $i \in \{0, 1\}$ telles que
  - la seule variable libre de $\phi(x_i)$ est x_i;
  - $\psi(x_i, x_{1-i})$ est une conjonction de formules atomiques de la forme $R(x, y)$ telle que pour au moins un conjoint $\{x, y\} = \{x_0, x_1\}$;

  alors $\forall x_i \ (\psi(x_i, x_{1-i}) \Rightarrow \phi(x_i))$ est dans WLGF2.

- WLGF2 est **PSPACE**-complet.

Exercice: traduction vers WLGF2

- WLGF2 admet des conjonctions dans la garde contrairement à GF.
- $\forall x \ R(x, x)$ n'appartient pas à GF $\cup$ WLGF2.
- Définir une traduction de T vers GF en utilisant le fait que $\Box\phi \Leftrightarrow \phi \wedge \Box\phi$ est T-valide. Caractériser le coût de la traduction.
- Définir une traduction polynomiale de T vers WLGF2.

Mécanisation des logiques modales

- Méthodes “directes”:
  - calculs analytiques: tableaux, séquents
  - résolution
  - automates

- Traduction vers
  - d’autres logiques modales (PDL, μ-calcul, ...)
  - des fragments de la logique classique (FO2, GF, ...)
  - des logiques du second-ordre (S2S, μLGF)

Algorithme de Ladner - Principe

- Construction d'un arbre fini d'ensemble de formules à partir duquel un modèle pour la formule ϕ peut être construit.
- Chaque ensemble de formules est un sous-ensemble d'un ensemble de formules de taille polynomial en la taille de ϕ Exemple: l'ensemble des sous-formules.
- Si un noeud n est étiqueté par X et $\Box\psi \not\in X$ avec $\Box\psi$ "sous-formule" de ϕ alors il existe un fils noeud n' de n étiqueté par X' tel que $\psi \not\in X'$.
- Pour assurer la terminaison, le degré modal des ensembles de formules peut décroître avec la profondeur dans l'arbre ou bien une détection de cycle est opérée.

Sous-formules

- **Définition .** (sous-formules) Soit X un ensemble fini de formules. On note $sub(X)$ l'ensemble des sous-formules de X défini comme le plus petit ensemble vérifiant les propriétés ci-dessous:
  - $X \subseteq sub(X)$,
  - si $\neg\psi \in sub(X)$ alors $\psi \in sub(X)$,
  - si $\Box\psi \in sub(X)$ alors $\psi \in sub(X)$,
  - si $\psi_1 \wedge \psi_2 \in sub(X)$ alors $\{\psi_1, \psi_2\} \subseteq sub(X)$.

  ∇

- $\mathrm{card}(sub(\{\phi\})) \leq |\phi|$.
- Un ensemble X est fermé si $X = sub(X)$.

Sous-formules (suite)

- **Définition .** Soit X un ensemble fini de formules. On définit une famille $sub(i, X)$, $i \in \mathbb{N}$, de sous-ensembles de $sub(X)$ comme la plus petite famille de sous-ensembles telle que:
  - $sub(0, X) = sub(X)$,
  - chaque ensemble $sub(i, X)$ est fermé,
  - si $\Box\psi \in sub(i, X)$, alors $\psi \in sub(i + 1, X)$.

∇

- **Lemme.** Pour i strictement supérieur au degré modal de ϕ, $sub(i, \{\phi\}) = \emptyset$.
- Dans le cas de logiques modales dont les opérateurs modaux sont indicés par des lettres d'un alphabet Σ, de façon analogue, des ensembles $sub(\sigma, X)$ sont définis avec $\sigma \in \Sigma^*$.

Consistence

Définition . (i-consistence) Soit $X \subseteq sub(i, \{\phi\})$ pour $i \geq 0$. X est dit i-consistent ssi pour $\psi \in sub(i, \{\phi\})$,

- si $\psi = \neg\varphi$ alors $\psi \in X$ ssi $\varphi \notin X$.
- si $\psi = \varphi_1 \wedge \varphi_2$ alors $\psi \in X$ ssi $\{\varphi_1, \varphi_2\} \subseteq X$.

∇

Définition . (relation d'accessibilité syntaxique $\prec$) Soient $X, Y \subseteq sub(\{\phi\})$. $X \prec Y \overset{\text{def}}{\Leftrightarrow}$ pour chaque $\Box\psi \in X$, on a $\psi \in Y$. ∇

Principe de l'algorithme

- $\mathtt{K} - \mathtt{WORLD}(\Sigma, i, \phi)$ admet comme arguments:
  - ϕ: formule dont on teste la satisfaisabilité.
  - $i \geq 0$: profondeur courante de l'arbre construit.
  - Σ une séquence finie de sous-ensembles de $sub(\{\phi\})$ avec pour $i > 1$, le dernier élément de Σ est un sous-ensemble de $sub(i, \{\phi\})$.
  - Σ contient l'historique d'une branche et on cherche à satisfaire le dernier élément de Σ.
- Nous allons montrer que ϕ est K-satisfaisable ssi il existe $X \subseteq sub(\{\phi\})$ tel que $\phi \in X$ et $\mathtt{K} - \mathtt{WORLD}(X, 0, \phi)$ retourne `Vrai`.
- Tous les sous-ensembles de $sub(\{\phi\})$ peuvent être énumérés avec un espace mémoire polynomial en la taille de ϕ.

Algorithme de Ladner

$\mathtt{K}-\mathtt{WORLD}(\Sigma, i, \phi)$

1. Si le dernier élément de Σ (noté $der(\Sigma)$) n'est pas i-consistent alors retourner `Faux`.
2. Pour $\Box\psi \in sub(i, \phi) \setminus der(\Sigma)$ faire

   pour chaque $X_\psi \subseteq sub(i+1, \phi) \setminus \{\psi\}$ tel que $der(\Sigma) \prec X_\psi$, appeler $\mathtt{K}-\mathtt{WORLD}(\Sigma \cdot X_\psi, i+1, \phi)$. Si tous les appels retournent `Faux` alors retourner `Faux`.
3. Retourner `Vrai`.

- Pour tout appel $\mathtt{K}-\mathtt{WORLD}(\Sigma, i, \phi)$ dans $\mathtt{K}-\mathtt{WORLD}(X, 0, \phi)$, $|\Sigma| = i+1$ et $i < |\phi|$.
- Le calcul de $\mathtt{K}-\mathtt{WORLD}(X, 0, \phi)$ ne nécessite qu'un espace mémoire polynomial en $|\phi|$ (Σ peut être implémenté comme une pile globale).

Correction de l'algorithme

- **Théorème.** Soit ϕ une formule modale. ϕ est K-satisfaisable ssi il existe $Y \subseteq sub(\{\phi\})$ tel que $\phi \in Y$ et $\mathtt{K} - \mathtt{WORLD}(Y, 0, \phi)$ retourne `Vrai`.
- Algorithme optimal dans le pire des cas car le problème de la satisfaisabilité pour K est **PSPACE**-complet.

Preuve

- Supposons que $\mathtt{K} - \mathtt{WORLD}(Y, 0, \phi)$ retourne `Vrai` pour $Y \subseteq sub(\{\phi\})$ tel que $\phi \in Y$
- Modèle $\mathcal{M} = \langle W, R, m \rangle$ et $w \in W$ tels que $\mathcal{M}, w \models \phi$.
- W est un sous-ensemble de $\mathcal{P}(sub(\{\phi\})) \times \{0, \ldots, |\phi|\}$.

Preuve (suite)

- $\langle X, i\rangle \in W$ ssi il existe une séquence $\langle \Sigma_1, i_1\rangle, \ldots, \langle \Sigma_k, i_k\rangle$ telle que
  1. pour $1 \leq j \leq k$, $\texttt{K}-\texttt{WORLD}(\Sigma_j, i_j, \phi)$ retourne `Vrai` et $\texttt{K}-\texttt{WORLD}(\Sigma_j, i_j, \phi)$ est appelé dans $\texttt{K}-\texttt{WORLD}(Y, 0, \phi)$.
  2. $\Sigma_1 = Y$, $i_1 = 0$, $der(\Sigma_k) = X$, $i_k = i$.
  3. $\texttt{K}-\texttt{WORLD}(\Sigma_j, i_j, \phi)$ appelle directement $\texttt{K}-\texttt{WORLD}(\Sigma_{j+1}, i_{j+1}, \phi)$.

- $\langle X, i\rangle R \langle X', i'\rangle \overset{\text{def}}{\Leftrightarrow}$ il existe un appel de $\texttt{K}-\texttt{WORLD}(\Sigma, i, \phi)$ qui retourne `Vrai` dans $\texttt{K}-\texttt{WORLD}(Y, 0, \phi)$ tel que:
  1. $der(\Sigma) = X$.
  2. $\texttt{K}-\texttt{WORLD}(\Sigma, i, \phi)$ appelle $\texttt{K}-\texttt{WORLD}(\Sigma', i', \phi)$ qui retourne `Vrai`.
  3. $der(\Sigma') = X'$.

- $v(p) = \{\langle X, i\rangle : p \in X\}$.

Preuve (suite)

- On a $\langle X, i\rangle R\langle X', i'\rangle$ implique $X \prec X'$.
- On peut montrer que pour $\langle X, i\rangle \in W$, pour $\psi \in sub(i, \{\phi\})$, $\psi \in X$ ssi $\mathcal{M}, \langle X, i\rangle \models \psi$.
- Donc $\mathcal{M}, \langle Y, 0\rangle \models \phi$.
- Supposons maintenant qu'il existe un modèle $\mathcal{M} = \langle W, R, v\rangle$ et $w \in W$ tels que $\mathcal{M}, w \models \phi$.
- On montre que pour $i \geq 0$, si Σ est une séquence finie de sous-ensembles telles que $der(\Sigma) \subseteq sub(i, \{\phi\})$ et s'il existe un modèle $\mathcal{M}' = \langle W', R', v'\rangle$ et $w' \in W'$ tels que pour $\psi \in sub(i, \{\phi\})$, $\mathcal{M}', w' \models \psi$ ssi $\psi \in X$, alors $\mathtt{K-WORLD}(\Sigma, i, \phi)$ retourne `Vrai`.
- En prenant $i = 0$ et $\Sigma = Y = \{\psi \in sub(\{\phi\}) : \mathcal{M}, w \models \psi\}$, on obtient que $\mathtt{K-WORLD}(Y, 0, \phi)$ retourne `Vrai`.
- La preuve est par récurrence sur i.

Algorithme de Ladner pour S4 (I)

- Dans la définition des ensembles $sub(i, \phi)$ on ajoute:
  si $\Box\psi \in sub(i, X)$, alors $\Box\psi \in sub(i+1, X)$.
  $sub(i, \{\phi\})$ n'est jamais vide si $\Box$ apparaît dans ϕ.

- Dans la définition de i-consistence on ajoute:
  si $\psi = \Box\varphi$ alors $\psi \in X$ implique $\varphi \in X$.

- La relation d'accessibilité syntaxique est définie ainsi pour $X, Y \subseteq sub(\{\phi\})$:
  $X \prec Y \overset{\text{def}}{\Leftrightarrow}$ pour chaque $\Box\psi \in X$, on a $\Box\psi, \psi \in Y$.

- La détection de cycles dans Σ devient nécessaire.

Algorithme de Ladner pour S4 (II)

$\mathtt{S4} - \mathtt{WORLD}(\Sigma, i, \phi)$

1. Si le dernier élément de Σ n'est pas i-consistent alors retourner `Faux`.
2. Pour $\Box\psi \in sub(i, \phi) \setminus der(\Sigma)$ faire

   S'il n'y a pas de X dans Σ tel que $\psi \notin X$ et $der(\Sigma) \prec X$ alors
   pour chaque $X_\psi \subseteq sub(i+1, \phi) \setminus \{\psi\}$ tel que $der(\Sigma) \prec X_\psi$, appeler $\mathtt{S4} - \mathtt{WORLD}(\Sigma \cdot X_\psi, i+1, \phi)$. Si tous les appels retournent `Faux` alors retourner `Faux`.
3. Retourner `Vrai`.

Majoration de la taille de Σ

Lemme. Pour $Y \subseteq sub(\phi)$, tout appel $\mathtt{S4} - \mathtt{WORLD}(\Sigma, i, \phi)$ dans $\mathtt{S4} - \mathtt{WORLD}(Y, 0, \phi)$ vérifie que $|\Sigma| \leq |\phi|^2$.

- $\Sigma = \Sigma_1 \cdots \Sigma_n$ où chaque Σ_i contient les mêmes $\Box$-formules et il existe une unique $\Box$-formule dans $\Sigma_{i+1} \setminus \Sigma_i$.
- $n \leq |\phi|$ et $(|\Sigma_{i+1}| - |\Sigma_{i+1}|)$ est borné par $|\phi|$.

Théorème. Soit ϕ une formule modale. ϕ est S4-satisfaisable ssi il existe $Y \subseteq sub(\{\phi\})$ tel que $\phi \in Y$ et $\mathtt{S4} - \mathtt{WORLD}(Y, 0, \phi)$ retourne `vrai`.

Corollaire. Le problème de la S4 satisfaisabilité est dans **PSPACE**.

Bilan sur l'algorithme de Ladner

- Obtention de bornes de complexité optimales pour K, S4 et pour bien d'autres logiques.

- Algorithmes très simples et facilement adaptables.

- Beaucoup de calculs inutiles à cause du choix des ensembles.

- **Exercice.** Définir un algorithme de Ladner pour la logique bimodale avec opérateurs $[1]$ et $[2]$ caracterisée par les modèles $\langle W, R_1, R_2, v\rangle$ tels que $R_1 \subseteq R_2$ et R_1, R_2 sont réflexives et transitives. En déduire que le problème de satisfaisabilité pour cette logique est **PSPACE**-complet.

Alternative: automates d'arbres

- Nombreuses réductions vers le problème de vacuité d'automates d'arbres (PDL, μ-calcul modal, etc.). ϕ est satisfaisable ssi $\mathrm{L}(\mathcal{A}_\phi) \neq \emptyset$.
- Propriété de la structure arborescente des modèles: pour toute formule satisfaisable il existe un arbre (éventuellement infini) à partir duquel un modèle peut être facilement construit.
- Traduction simple mais on prend avantage des optimisations des algorithmes sur les automates d'arbres.
- L'algorithme de Ladner est très similaire au test du vide pour certains automates d'arbres.
- On traite la construction de tels automates pour S4.

Rappel

- **Définition .** (consistence) Soit $X \subseteq sub(\phi)$. X est dit maximallement consistent pour S4 $\overset{\text{def}}{\Leftrightarrow}$ pour $\psi \in sub(\phi)$,
  - si $\psi = \neg\varphi$ alors $\psi \in X$ ssi $\varphi \notin X$.
  - si $\psi = \varphi_1 \wedge \varphi_2$ alors $\psi \in X$ ssi $\{\varphi_1, \varphi_2\} \subseteq X$.
  - si $\psi = \Box\varphi$ alors $\psi \in X$ implique $\varphi \in X$.

∇

- Relation d'accessibilité syntaxique: pour $X, Y \subseteq sub(\phi)$, $X \prec Y$ $\overset{\text{def}}{\Leftrightarrow}$ pour chaque $\Box\psi \in X$, on a $\Box\psi, \psi \in Y$.

- Σ, K-arbre $\mathcal{T}$: $\{1, \ldots, K\}^* \to \Sigma$
  Σ est un alphabet fini, $\{1, \ldots, K\}^*$ est un arbre infini.

Arbres d'Hintikka

- Un arbre de Hintikka $\mathcal{T}$ pour ϕ est un $\mathcal{P}(sub(\phi)), |\phi|$-arbre ($K = |\phi|$) tel que $\phi \in \mathcal{T}(\epsilon)$ et pour $s \in \{1, \ldots, K\}^*$,
  - $\mathcal{T}(s)$ est maximallement consistent.
  - si $\Box\psi \in sub(\phi) \setminus \mathcal{T}(s)$ alors il existe $i \in \{1, \ldots, K\}$ tel que $\psi \notin \mathcal{T}(s \cdot i)$.
  - $\mathcal{T}(s) \prec \mathcal{T}(s \cdot i)$ pour $i \in \{1, \ldots, K\}$.

- **Lemme.** ϕ est S4 satisfaisable ssi ϕ possède un arbre de Hintikka.
  Preuve laissée en exercice.

Automates d'arbre de Büchi

- Les arbres de Hintikka pour ϕ sont construits avec des automates de Büchi.
- **Définition .** Un automate de Büchi $\mathcal{A} = \langle \Sigma, Q, \delta, I, F \rangle$ pour Σ, K-arbres est un modèle opérationnel tel que
  - Q est un ensemble fini d'états,
  - Σ est un alphabet fini,
  - $\delta \subseteq Q \times \Sigma \times Q^K$ est la relation de transition,
  - $I \subseteq Q$ est l'ensemble des états initiaux,
  - $F \subseteq Q$ est l'ensemble des états finaux.

∇

- Une exécution r sur un $\Sigma - K$-arbre $\mathcal{T}$ est un Q, K-arbre tel que $r(\epsilon) \in I$ et pour $s \in \{1, \ldots, K\}^*$,

$$\langle r(s), \mathcal{T}(s), r(s \cdot 1), \ldots r(s \cdot K) \rangle \in \delta$$

Automates d'arbre de Büchi (suite)

- Une exécution r est acceptante $\overset{\text{def}}{\Leftrightarrow}$ sur chaque branche de r il existe un état de F qui apparaît infiniment.
- $\mathrm{L}(\mathcal{A})$: ensemble des Σ, K-arbres qui admettent une exécution acceptante.
- Tester le vide de $\mathrm{L}(\mathcal{A})$ est polynomial en $|\mathcal{A}| + K$
  En fait test en temps $\mathcal{O}(|\delta|^2)$.

Automate $\mathcal{A}_\phi$

$\mathcal{A}_\phi = \langle \Sigma, Q, \delta, I, F \rangle$ avec

- $Q = \Sigma$: ensemble des ensembles maximallement consistents pour ϕ.
- $I = \{X \in Q : \phi \in X\}$, $F = Q$.
- $\langle q', X, q'_1, \ldots, q'_K \rangle \in \delta$ ssi
  - $q' = X$ et $q' \prec q'_i$ pour $i \in \{1, \ldots, K\}$.
  - si $\Box\psi \in sub(\phi) \setminus q'$ alors il existe $i \in \{1, \ldots, K\}$ tel que $\psi \notin q'_i$.

Propriétés de $\mathcal{A}_\phi$

- $|\mathcal{A}_\phi|$ est exponentielle en $|\phi|$.
- Tester si $\mathrm{L}(\mathcal{A}_\phi) \neq \emptyset$ demande un temps exponentiel en $|\phi|$.
- **Lemme.** $\mathrm{L}(\mathcal{A}_\phi)$ est exactement l'ensemble des arbres de Hintikka pour ϕ.
- **Corollaire.** Une formule ϕ est S4-satisfaisable ssi $\mathrm{L}(\mathcal{A}_\phi) \neq \emptyset$.
- On n'obtient pas la borne **PSPACE**. Technique plutôt optimale pour les logiques **EXPTIME**-complètes.

Exercice

- A l'aide d'automates d'arbres, résoudre le problème de la satisfaisabilité pour la logique bimodale avec opérateurs $[1]$ et $[2]$ caracterisée par les modèles $\langle W, R_1, R_2, v\rangle$ tels que $R_2 = W \times W$ (R_1 est quelconque).

- En déduire que le problème de satisfaisabilité pour cette logique est dans **EXPTIME**. (En fait la **EXPTIME**-dureté est aussi vraie.)

Méthode des tableaux

- Méthode analytique: les règles du calcul se basent sur la décomposition syntaxique de la formule.
- Méthode développée par Smullyan pour la logique classique (vers 1968).
- Relation étroite avec les calculs de séquents de Gentzen.
- Utilisation éventuelle d’étiquettes dans le calcul
  - interprétées comme des états du modèle en cours de construction
  - utilisées comme structures de contrôle
- Règles plus fines que les notions de consistence maximale de l’algorithme de Ladner ou des constructions d’automates.

Quelques autres problèmes

- Problème de la $\mathcal{C}$-validité globale:

  entrée: un ensemble fini X de formules et une formule ϕ

  sortie: 1 si pour tous les modèles $\mathcal{M} = \langle W, R, v \rangle \in \mathcal{C}$ tels que pour $w \in W$ et $\psi \in X$, $\mathcal{M}, w \models \psi$, il existe $w_0 \in W$ tel que $\mathcal{M}, w_0 \models \phi$ (noté $X \models_l \phi$).

  (il existe d'autres variantes)

- Problème de la $\mathcal{C}$-satisfaisabilité globale:

  entrée: un ensemble fini X de formules et une formule ϕ

  sortie: 1 s'il existe $\mathcal{M} = \langle W, R, v \rangle \in \mathcal{C}$ et $w \in W$ tel que pour $w \in W$ et $\psi \in X$, $\mathcal{M}, w \models \psi$ et il existe $w_0 \in W$ tel que $\mathcal{M}, w_0 \models \phi$.

  (il existe d'autres variantes)

Calculs des tableaux de F. Massacci

- La méthode “Single Step Tableaux” de Fabio Massacci (vers 2000) raffinent la méthode des tableaux de Melvin Fitting (introduite vers 1980).
- Approche modulaire: on ajoute de nouvelles règles si R est de plus en plus contraint dans les modèles.
- Propriétés algorithmiques fortes: on retrouve les bornes de complexité optimales en appliquant les règles avec une stratégie simple.
- Méthode des tableaux présentée pour les logiques suivantes:
  - K, S4
  - S5 (R est une relation d’équivalence)
  - B (R est réflexive et symétrique)

Tableaux avec formules préfixées

- Préfixe σ: séquence finie non vide d'entiers
  - σ sera interprété comme un état d'un modèle
  - concaténation: $\sigma \cdot \sigma'$.
- Formule préfixée: paire de la forme $\sigma : \psi$ où σ est un préfixe et ψ est une formule (on se restreint à $\neg, \wedge, \Box$).
  - $\sigma : \psi$ présent dans une preuve implique que l'interprétation de σ vérifie la formule ψ.
- Tableaux: arbre dont les noeuds sont étiquetés par des formules préfixées.
- Branche: notion standard dans un arbre.
- σ est présent sur une branche $\overset{\text{def}}{\Leftrightarrow}$ une formule préfixée de la forme $\sigma : \psi$ étiquette un noeud de la branche.

Propriétés fondamentales du calcul (I)

- La racine est étiquetée par $1 : \neg\phi$ où (les entrées du problème sont ϕ et X) et les règles du calcul permettent de construire des tableaux à partir de la racine.
- On recherche à construire un tableau qui corresponde à un modèle pour $\neg\phi$.
- Les règles d'inférence sont analytiques et si $\sigma \cdot n$ est un nouveau préfixe présent sur la branche alors nécessairement σ était déjà sur la branche.
- En conséquence, l'ensemble des préfixes présents sur une branche forme un arbre.
- La différence de longueur entre deux préfixes d'une même règle est au plus 1 ("Single Step Tableaux").

Propriétés fondamentales du calcul (II)

- Pour toute logique $\mathcal{L}$ parmi K, B, S4 et S5 il existe un opérateur de fermeture $\mathcal{C}_{\mathcal{L}}$ tel que pour tout modèle $\langle W, R, v\rangle$, $\langle W, \mathcal{C}_{\mathcal{L}}(W, R), v\rangle$ est un modèle de $\mathcal{L}$:
  - $\mathcal{C}_{\mathrm{K}}$ est l'identité,
  - $\mathcal{C}_{\mathrm{S4}}$ est la fermeture réflexive et transitive.

- Soit $\rhd^*$ la fermeture de la relation "préfixe immédiat" pour l'ensemble des préfixes présents sur une branche.

- Exemple pour S4: $\sigma \rhd^* \sigma'$ ssi σ est un préfixe de σ'.

- Si $\sigma : \Box\psi$ et σ' sont présents sur une branche avec $\sigma \rhd^* \sigma'$ alors il est possible d'étendre cette branche de façon à avoir $\sigma' : \psi$ sur la branche ainsi étendue.

Règles du calcul propositionnel

$$\frac{\sigma : \psi \wedge \psi'}{\begin{array}{c}\sigma : \psi \\ \sigma : \psi'\end{array}} \ (\alpha)$$

$$\frac{\sigma : \neg(\psi \wedge \psi')}{\sigma : \neg\psi \ \mid \ \sigma : \neg\psi'} \ (\beta)$$

$$\frac{\sigma : \neg\neg\psi}{\sigma : \psi} \ (\neg\neg)$$

Règles modales communes

$$\frac{\vdots}{\sigma : \psi} \ \sigma \text{ présent sur la branche et } \psi \in X$$

$$\frac{\sigma : \neg\Box\psi}{\sigma \cdot n : \neg\psi} \ (\pi) \quad \sigma \cdot n \text{ nouveau sur la branche}$$

Règles propres à chaque logique (I)

- Règle pour les logiques K, B, S4 et S5:

$$\frac{\sigma : \Box\psi}{\sigma \cdot n : \psi}\ (\mathrm{K})$$

σ et $\sigma \cdot n$ sont présents sur la branche.

- Règle pour les logiques B, S4 et S5:

$$\frac{\sigma : \Box\psi}{\sigma : \psi}\ (\mathrm{T})$$

- Règle pour les logiques S4 et S5:

$$\frac{\sigma : \Box\psi}{\sigma \cdot n : \Box\psi}\ (4)$$

σ et $\sigma \cdot n$ sont présents sur la branche.

Règles propres à chaque logique (II)

- Règle pour la logique S5:

$$\frac{\sigma \cdot n : \Box\psi}{\sigma : \Box\psi}\ (4^{\mathrm{R}})$$

σ et $\sigma \cdot n$ sont présents sur la branche.

- Règle pour la logique B:

$$\frac{\sigma \cdot n : \Box\psi}{\sigma : \psi}\ (\mathrm{B})$$

σ et $\sigma \cdot n$ sont présents sur la branche.

- (K) peut être simulée avec (4) et (T):

$$\cfrac{\cfrac{\sigma : \Box\psi}{\sigma \cdot n : \Box\psi}\ (4)}{\sigma \cdot n : \psi}\ (\mathrm{T})$$

Preuve par tableau

- Une branche est fermée s'il existe une formule préfixée $\sigma : \psi$ telle que $\sigma : \psi$ et $\sigma : \neg\psi$ sont sur la branche.

- Un tableau est fermé si chaque branche est fermée.

- Une preuve par tableau pour la formule ϕ avec ensemble d'hypothèses globales X est un tableau fermé dont la racine est $1 : \neg\phi$.

- **Théorème.** Pour les logiques K, B, S4 et S5, $X \models_l \phi$ ssi il existe une preuve par tableau de ϕ avec ensemble d'hypothèses globales X.

Preuve de $\Box p \Rightarrow \Box\Box p$ *pour S4*

$$1 : \neg(\Box p \Rightarrow \Box\Box p)$$
$$1 : \Box p$$
$$1 : \neg\Box\Box p$$
$$1 \cdot 1 : \neg\Box p$$
$$1 \cdot 1 : \Box p$$

- $\Box p \Rightarrow \Box\Box p$ est S4 valide.

Preuve de correction (I)

- Soit $\mathcal{B}$ une branche et $\mathcal{M} = \langle W, R, v \rangle$ un modèle de la logique $\mathcal{L}$. Une interprétation f est une fonction de l'ensemble des préfixes de $\mathcal{B}$ vers W telle que $f(\sigma) R f(\sigma \cdot n)$ pour tous les σ et $\sigma \cdot n$ présents sur la branche.

- Une branche $\mathcal{B}$ est dite réalisable par rapport à un ensemble X d'hypothèses globales ssi il existe un $\mathcal{L}$-modèle $\mathcal{M} = \langle W, R, v \rangle$ et une interprétation f tels que
  - pour tous les $w \in W$ et $\psi \in X$, $\mathcal{M}, w \models \psi$.
  - pour $\sigma : \psi$ présent sur $\mathcal{B}$, on a $\mathcal{M}, f(\sigma) \models \psi$.

- Un tableau est réalisable s'il contient une branche réalisable.

Preuve de correction (II)

- **Lemme.** Si un tableau $\mathcal{T}$ est réalisable alors tous les tableaux $\mathcal{T}'$ obtenus par application d'une régle sont aussi réalisables.

  La preuve est facile.

- **Corollaire.** Pour les logiques K, B, S4 et S5, s'il existe une preuve par tableau de ϕ avec pour ensemble d'hypothèses globales X alors $X \models_l \phi$.

Complétion d'une branche

- Une formule préfixée $\sigma : \psi$ est réduite pour la règle (r) sur une branche $\mathcal{B}$
  - si (r) est de la forme

$$\frac{\sigma : \psi}{\sigma' : \psi'}\ (r)$$

    alors $\sigma' : \psi'$ est sur $\mathcal{B}$.

  - si (r) est de la forme

$$\frac{\sigma : \psi}{\sigma' : \psi' \ \mid \ \sigma'' : \psi''}\ (r)$$

    alors soit $\sigma' : \psi'$ soit $\sigma'' : \psi''$ est sur $\mathcal{B}$.

- Une formule préfixée $\sigma : \psi$ est complétement réduite sur une branche si elle est réduite pour toutes les règles.

Complétion d'une branche (suite)

- Un préfixe σ est complétement réduit si toutes les formules de la forme $\sigma : \psi$ sont complétement réduites.
- Une branche $\mathcal{B}$ est complétée si tous les préfixes présents sur la branche sont complétement réduits.
- Une branche est ouverte si elle est complétée mais n'est pas fermée.
- Un tableau est ouvert s'il contient une branche ouverte.

Tableau ouvert pour $\neg\Box p \wedge \neg\Box\neg p$

$$1 : \neg\Box p \wedge \neg\Box\neg p$$
$$1 : \neg\Box p$$
$$1 : \neg\Box\neg p$$
$$1 \cdot 1 : \neg\neg p$$
$$1 \cdot 1 : p$$
$$1 \cdot 2 : \neg p$$

- $\neg\Box p \wedge \neg\Box\neg p$ est satisfaisable pour les logiques K, B, S4 et S5.

Preuve de complétude (quelques idées)

- Définition d'une procédure systématique équitable pour appliquer les règles.
- Une branche ouverte avec $1 : \neg\phi$ à la racine, permet de construire un contre-modèle pour ϕ.
- La procédure systématique garantit que si $\sigma : \Box\psi$ et σ' sont présents sur une branche avec $\sigma \rhd^* \sigma'$ alors il est possible d'étendre cette branche de façon à avoir $\sigma' : \psi$ sur celle-ci.
- $\rhd^*$ vérifie les propriétés des relations d'accessibilité des modèles de la logique.

Procédures de décision

- La complétude n'assure pas en soi l'existence d'une stratégie complète qui termine.

- On s'intéresse à la satisfaisabilité avec l'ensemble des hypothèses globales vide.

- ϕ est en forme normale négative avec les opérateurs $\vee, \wedge, \neg, \Box, \Diamond$: $\neg$ n'est autorisé que devant une variable propositionnelle.

- Toute formule modale construite avec $\neg, \wedge, \Box$ admet une formule équivalente en forme normale négative (construction en temps polynomial).

Deux techniques élémentaires

- Nouvelles règles:

$$\frac{\sigma : \psi \vee \psi'}{\sigma : \psi \mid \sigma : \psi'}\ (\beta) \quad \frac{\sigma : \Diamond\psi}{\sigma \cdot n : \psi}\ (\pi) \quad \sigma \cdot n \text{ nouveau sur la branche}$$

- **Intuition**: Pour chaque logique $\mathcal{L}$, il existe une borne de longueur $hb_{\mathcal{L}}$ sur les préfixes d'une branche telle que après cette borne soit il n'y a plus d'opérateurs modaux ou bien les formules se répètent avec des préfixes plus longs.
- Pour la procédure de décision deux techniques sont utilisées:
  - **Non-redondance des inférences**: Appliquer une règle (r) avec pour prémisse $\sigma : \psi$ seulement si cette formule n'est pas déjà réduite pour (r) sur la branche.
  - **Restriction sur les préfixes**: L'application de la π-règle est restreinte aux préfixes de longueur inférieure à $hb_{\mathcal{L}}$.

Bornes de hauteur

- d: degré modal de $\neg\phi$.
- n: nombre de formules de la forme $\Box\psi$ dans $\neg\phi$.
- p: nombre de formules de la forme $\Diamond\psi$ dans $\neg\phi$.
- d_p: imbrication maximale d'opérateurs $\Diamond$ qui ne sont dans la portée d'aucun opérateur $\Box$.
- $hb_{\mathrm{K}} = hb_{\mathrm{B}} = 1 + d$, $hb_{\mathrm{S4}} = 2 + d_p + p \times n$.
- **Lemme.** Pour $\mathcal{L}$ parmi K, B et S4, toute stratégie qui utilise la restriction sur les préfixes et la non-redondance des inférences, garantit la terminaison de la construction des tableaux à partir d'une racine de la forme $1 : \neg\phi$.
  (La preuve est facile.)

Quelques définitions

- Sur une branche $\mathcal{B}$, σ_0 est une copie de σ ssi pour toute formule ψ, $\sigma : \psi$ est sur $\mathcal{B}$ ssi $\sigma_0 : \psi$ est sur $\mathcal{B}$.
- Un préfixe est π-réduit sur une branche $\mathcal{B}$ s'il est réduit pour toutes les règles sauf pour la π-règle.
- Une branche $\mathcal{B}$ est π-complétée si
  - tous les préfixes sont π-réduits sur $\mathcal{B}$ et
  - pour chaque préfixe σ non complétement réduit, il existe une copie σ_0 de σ plus courte que σ qui soit complétement réduite.
- Une branche $\mathcal{B}$ est π-complétée modalement si
  - tous les préfixes sont π-réduits sur $\mathcal{B}$ et
  - pour chaque formule $\sigma : \Diamond\psi$ non réduite, il existe une copie σ_0 de σ plus courte que σ telle que $\sigma_0 : \Diamond\psi$ soit réduite.

Terminaison

Théorème. Pour $\mathcal{L}$ parmi K, B et S4, toute stratégie qui utilise la restriction sur les préfixes et la non-redondance des inférences, assure la terminaison à partir de $1 : \neg\phi$ dans un des cas suivants:

- une preuve par tableau est trouvée.
- sur chaque branche, une règle est applicable.
- au moins une branche peut être réduite à une branche π-complétée (à une branche π-complétée modalement pour S4).

La preuve est facile pour K et B mais plus difficile pour S4.

Exercice

- Les modèles de la logique $\mathcal{L}^\alpha$ ($\alpha \geq 2$) vérifient que $\overbrace{R \circ \ldots \circ R}^{\alpha \text{ fois}} \subseteq R$ et R est réflexive.
- $\mathcal{L}^2$ correspond à S4.
- Définir un calcul pour $\mathcal{L}^\alpha$ ($\alpha \geq 2$) qui vérifie la propriété de la sous-formule mais la différence de longueur entre deux préfixes d'une même règle peut être supérieure à 1.
- Montrer la correction du calcul.

Calculs de séquents

- Séquents: paire $\Gamma \vdash \Delta$ de multi-ensembles finis composés de formules modales.
- Γ, ψ dénote le multi-ensemble $\Gamma \cup \{\psi\}$ avec l'opérateur d'union $\cup$ pour les multi-ensembles.
- $\Box\Gamma \stackrel{\text{def}}{=} \bigcup_{\psi \in \Gamma} \Box\psi$.
- Preuve: Arbre fini dont les noeuds sont étiquetés par des séquents tel que
  - les feuilles sont étiquetées par des axiomes: dans la suite les axiomes sont de la forme $\Gamma, \psi \vdash \Delta, \psi$,
  - les contraintes entre les noeuds fils et parents sont déterminées par les règles d'inférence.
- Une preuve de $\emptyset \vdash \phi$ est équivalente à la validité de ϕ.

Calcul des séquents pour K

- Axiomes: $\Gamma, \psi \vdash \Delta, \psi$.

- Règles propositionnelles:

$$\frac{\Gamma \vdash \Delta, \phi}{\Gamma, \neg\phi \vdash \Delta}\ (\neg \vdash) \qquad \frac{\Gamma, \phi \vdash \Delta}{\Gamma \vdash \Delta, \neg\phi}\ (\vdash \neg)$$

$$\frac{\Gamma, \phi_1, \phi_2 \vdash \Delta}{\Gamma, \phi_1 \wedge \phi_2 \vdash \Delta}\ (\wedge \vdash) \qquad \frac{\Gamma \vdash \Delta, \phi_1 \quad \Gamma \vdash \Delta, \phi_2}{\Gamma \vdash \Delta, \phi_1 \wedge \phi_2}\ (\vdash \wedge)$$

- Règle modale:

$$\frac{\Sigma \vdash \phi}{\Gamma, \Box\Sigma \vdash \Box\phi, \Delta}\ (\vdash \Box)$$

Propriétés du calcul

- Les règles structurelles sont réduites au minimum dans ce calcul.
- L'application des règles décroît strictement la taille des séquents (de bas en haut pour la recherche de preuve).
- Si $\Gamma \vdash \Delta$ a une preuve alors il existe une preuve de profondeur au plus de la somme des tailles de Γ et Δ.
- Déterminer si $\Gamma \vdash \Delta$ a une preuve peut être vérifié en espace polynomial en utilisant du backtracking à cause de la règle $(\vdash \Box)$.
- **Théorème.** $\Gamma \vdash \Delta$ a une preuve ssi $(\bigwedge_{\psi \in \Gamma} \psi) \;\Rightarrow\; (\bigvee_{\psi \in \Delta} \psi)$ est K-valide.

Relation avec les tableaux (I)

- Chercher une preuve pour $\Gamma \vdash \Delta$ équivaut à
  - déterminer la validité de $(\bigwedge_{\psi \in \Gamma} \psi) \;\Rightarrow\; (\bigvee_{\psi \in \Delta} \psi)$.
  - déterminer la non-satisfaisabilité de $(\bigwedge_{\psi \in \Gamma} \psi) \;\wedge\; (\bigwedge_{\psi \in \Delta} \neg\psi)$.

- Au séquent $\Gamma \vdash \Delta$ correspond une branche d'un tableau contenant les formules de Γ et $\neg\psi$ pour $\psi \in \Delta$.

- Correspondance entre les axiomes et la notion de fermeture dans les tableaux.

- La correspondance s'étend aux règles d'inférence équipées de stratégies.

- Pas de backtracking nécessaire avec les tableaux préfixés.

Relation avec les tableaux (II)

- La règle $(\vdash \Box)$ correspond aux inférences suivantes dans les tableaux:

$$\frac{\{\psi_1, \ldots, \psi_m\} \vdash \phi}{\Gamma, \Box\{\psi_1, \ldots, \psi_m\} \vdash \Box\phi, \Delta}$$

$$\begin{array}{l}
\vdots \\
\sigma : \Box\psi_j \\
\vdots \\
\sigma : \Box\psi_{j'} \\
\vdots \\
\sigma : \neg\Box\phi \\
\sigma \cdot n : \neg\phi \\
\sigma \cdot n : \psi_1 \\
\vdots \\
\sigma \cdot n : \psi_m
\end{array}$$

Calcul des séquents pour S4

- Ajout de la règle $(\Box\vdash)$:

$$\frac{\Gamma, \Box\psi, \psi \vdash \Delta}{\Gamma, \Box\psi \vdash \Delta} \ (\Box\vdash)$$

- La règle $(\vdash\Box)$ est remplacée par:

$$\frac{\Box\Sigma \vdash \phi}{\Gamma, \Box\Sigma \vdash \Box\phi, \Delta} \ (\vdash\Box)$$

- La taille des séquents ne décroît plus strictement après application des règles.

- **Théorème.** $\Gamma \vdash \Delta$ a une preuve dans le calcul pour S4 ssi $(\bigwedge_{\psi\in\Gamma}\psi) \Rightarrow (\bigvee_{\psi\in\Delta}\psi)$ est S4-valide.

Quelques références (I)

- N. Alechina and S. Demri and M. de Rijke. A modal perspective on path constraints. Journal of Logic and Computation 13(9):939–956, 2003.
- H. Andreka and I. Nemeti and J. van Benthem. Modal languages and Bounded Fragments of Predicate Logic. Journal of Philosophical Logic 27(3): 217–274, 1998.
- P. Blackburn, M. de Rijke, Y. Venema. Modal logic, Cambridge University Press, 2001.
- M. Fitting. Proof methods for Modal and Intuitionistic Logics, Reidel, 1983.
- R. Goré. Tableaux method for modal and temporal logics. In Handbook of Tableaux Methods, 297–396, Kluwer, 1999.

Quelques références (II)

- R. Ladner. The computational complexity of provability in systems of modal propositional logic. SIAM Journal of Computing 6(3): 467–480, 1977.
- F. Massacci. Single Step Tableaux for Modal Logics: Computational Properties, Complexity and Methodology. Journal of Automated Reasoning 24: 319–364, 2000.
- M. Vardi. Why is modal logic so robustly decidable? In DIMACS Series in Discrete Mathematics and Theoretical Computer Science 31, American Mathematical Society, 149–183.
- M. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. Journal of Computer and System Sciences 32: 183–221, 1986.