

Mathématiques pour l'Informatique

Logique

David Teller

13/03/2007



Introduction

Booléens

Logique formelle

Logique propositionnelle

Interprétations

Séquents

Logiques

Termes de preuves

Preuves et programmes

Logique des prédicats

Conclusions



Au programme

"La logique est [...] l'étude des règles formelles que doit respecter toute déduction correcte."

Accessoirement, "Algorithmique = Logique + Contrôle."

- ▶ Booléens et algèbres de Boole.
- ▶ Logique formelle.
- ▶ Déductions formelles.
- ▶ ... vers la déduction automatique.

Introduction

Booléens

Logique formelle

Logique propositionnelle

Interprétations

Séquents

Logiques

Termes de preuves

Preuves et programmes

Logique des prédicats

Conclusions



Intuitivement

Definition (Algèbre minimale)

L'*algèbre minimale de Boole*, notée \mathbf{B} est l'ensemble $\{\text{Vrai}, \text{Faux}\}$ doté des lois \wedge (conjonction) et \neg (négation) définies par

a	b	$a \wedge b$	$\neg a$
V	V	V	F
V	F	F	F
F	V	F	V
F	F	F	V

À partir de \wedge et \neg , on définit \vee .



Algèbre de Boole

Prêts ?

Definition (Algèbre de Boole)

Une *algèbre de Boole* est un ensemble E doté de deux éléments distincts \perp et \top , de deux opérations binaires sur E \sqcup et \sqcap et d'une opération unaire sur E $\bar{}$ – telles que

- ▶ \sqcup et \sqcap sont idempotentes, associatives, commutatives, distributives l'une par rapport à l'autre
- ▶ $\forall x, y \in E, x \sqcap (x \sqcup y) = x = x \sqcup (x \sqcap y)$
- ▶ \perp est neutre pour \sqcup et absorbante pour \sqcap
- ▶ \top est neutre pour \sqcap et absorbante pour \sqcup
- ▶ $x \sqcap \bar{x} = \perp$
- ▶ $x \sqcup \bar{x} = \top$.



Exemples

Proposition

L'*algèbre de Boole minimale* $(\{F, V\}, \vee, \wedge, \neg)$ est une *algèbre de Boole* avec $F = \perp$ et $V = \top$ (ouf !)

Proposition

L'*algèbre de Boole minimale prise à l'envers* $(\{F, V\}, \wedge, \vee, \neg)$ avec $V = \perp$ et $F = \top$ est encore une *algèbre de Boole* !



Exemples

Proposition

Si E n'est pas vide, l'ensemble des parties de E forme une algèbre de Boole minimale.

Preuve Il suffit de prendre $\perp = \emptyset$, $\top = E$, $\cap = \cap$, $\cup = \cup$ et le complément dans E pour $-$.

...ou $\top = \emptyset$, $\perp = E$, $\cap = \cup$, $\cup = \cap$ et le complément dans E pour $-$!

Q Contrairement à \mathbf{B} , $\mathcal{P}(E)$ contient plus de deux éléments. Comment l'interpréter ?

Typiquement, on interprète E comme un ensemble de variables et $X \subseteq E$ comme l'ensemble des variables dont la valeur est Vrai.

Note Ce qui précède était pour votre culture générale. Vous ne vous en servirez pas souvent (voire jamais).



Introduction

Booléens

Logique formelle

Logique propositionnelle

Interprétations

Séquents

Logiques

Termes de preuves

Preuves et programmes

Logique des prédicats

Conclusions



Logique formelle

La logique formelle provient de tentatives de comprendre le raisonnement humain.

En tant qu'outil de psychologie, ça a ses limites, mais ça reste la meilleure manière de comprendre ce qu'est une démonstration correcte – et, au juste, ce qu'elle prouve.

Note Même si c'est tentant, il n'est généralement pas possible d'écrire une preuve un peu complexe entièrement en logique formelle. Pour faire ce genre de choses, on utilise Coq...

Propositions

Definition (Proposition informelle)

Une *proposition informelle* est une phrase avec une signification précise et constante dans le temps, et qui est forcément vraie ou fausse – même si on ne sait pas encore dans lequel des deux cas on est.

Exemples “ $1 = 2$ ”, “ $\forall n \in \mathbf{N} \setminus \{0, 1, 2\}, \forall a, b, c \in \mathbf{N}^*, a^n + b^n \neq c^n$ ”, “il a gelé quelque part à Bourges dans la nuit du 19 au 20 mars 2007”...



Implication

Definition (Implication)

Si P et Q sont deux propositions traitant des mêmes objets, on dit que P *implique* Q (ou $P \Rightarrow Q$), si soit Q est vraie, soit P est fausse.

Definition (Implication)

Si P et Q sont deux propositions, la *réciproque* de $P \Rightarrow Q$ est $Q \Rightarrow P$.

Note La validité de la réciproque est totalement indépendante de la validité du sens direct.

Proposition

L'implication est une relation d'ordre.



Et à partir de là ?

Tout dépend de ce que nous souhaitons faire avec la logique...

Logique propositionnelle Logique traitant des relations entre symboles et entre preuves.

Logique des prédicats Logique propositionnelle + quantificateurs.

Logique épistémique Logique des prédicats + une notion d'agents.

Logique linéaire Logique des prédicats + une notion de *coût* d'une hypothèse..



Introduction

Booléens

Logique formelle

Logique propositionnelle

Interprétations

Séquents

Logiques

Termes de preuves

Preuves et programmes

Logique des prédicats

Conclusions



Quelques symboles

Pour la suite, nous considérerons $\mathcal{N} = \{p, q, r \dots\}$, l'ensemble (dénombrable) des noms de variables propositionnelles.

Definition (Formule)

L'ensemble \mathcal{F} des *formules propositionnelles* est le sous-ensemble du monoïde libre engendré par $\mathcal{N} \cup \{\text{NON}, \text{IMPLIQUE}, \text{,}(\text{,})\}$ et défini par

$$\begin{aligned} \mathcal{F} ::= & \mathcal{N} \\ & | \text{NON } \mathcal{F} \\ & | \mathcal{F} \text{ IMPLIQUE } \mathcal{F} \end{aligned}$$

Note IMPLIQUE, \Rightarrow et les parenthèses sont, pour le moment, des symboles sans signification. En particulier, les notions de *vrai* et *faux* n'ont pas encore de sens !



Sémantique

Definition (Sémantique)

Dans n'importe quel langage, la *sémantique* d'une expression est la signification donnée à cette expression.

On définit toujours un langage par rapport à un langage plus simple. Ici, la sémantique d'une formule sera la détermination de sa vérité.



Validité

Definition (Interprétation)

L'*interprétation* des variables est une fonction $I : \mathcal{N} \mapsto \{V, F\}$ qui détermine quelles variables sont vraies.

Definition (Interprétation)

L'*interprétation* des formules à partir d'une interprétation I sur les variables est une fonction $J_I : \mathcal{F} \mapsto \{V, F\}$ définie par

- ▶ $\forall n \in \mathcal{N}, J_I(n) = I(n)$ est une interprétation de variables ;
- ▶ $J_I(\text{NON } f) = \overline{J_I(f)}$ (la négation dans $\{V, F\}$)
- ▶ $J_I(f \text{ IMPLIQUE } f') = J_I(f) \Rightarrow J_I(f')$.



Notes

Les symboles NON et IMPLIQUE font partie de la *formule*, tandis que \Rightarrow, \neg, V et F font partie du monde des mathématiques. En d'autres termes, nous calculons le *résultat* de la formule.

```
# let rec interprete_formule i = function
  | Non n -> I n
  | Neg f -> not (interprete_formule i f)
  | Implique (f, g) -> (interprete_formule i g) ||
    (not (interprete_formule i f))::
val interprete_formule : (string -> bool) -> formule -> bool = <f>
```



Tautologies & co.

Definition (Tautologie)

Une formule f est une tautologie si toutes ses interprétations sont vraies.

Definition (Satisfaisable)

Une formule f est *satisfaisable* s'il existe au moins une interprétation de f qui soit vraie.

Definition (Insatisfaisable)

Une formule f est *satisfaisable* si toutes ses interprétations sont fausses.

Définitions

Definition (Disjonction)

Pour tout f et tout f' , définissons la formule f OU f' (ou "disjonction de f et f' ") comme $\text{NON } f \text{ IMPLIQUE } f'$.

Definition (Conjonction)

Pour tout f et tout f' , définissons la formule f ET f' (ou "conjonction de f et f' ") comme $\text{NON } (f \text{ IMPLIQUE NON } f')$.

Quelques exemples

Proposition

Les formules p IMPLIQUE p , p IMPLIQUE q , p IMPLIQUE NON p sont-elles des tautologies? Satisfaisables? Insatisfaisables?

Encore des exemples

Proposition

Les formules p ET NON p , p OU NON p sont-elles des tautologies? Satisfaisables? Insatisfaisables?

- ▶ p ET NON p est insatisfaisable – pour ce faire, il suffit de vérifier avec toutes les interprétations de variables I telles que $I(p) = V$ et avec toutes les interprétations de variables I telles que $I(p) = F$
- ▶ p OU NON p est une tautologie – pour ce faire, il suffit de vérifier avec toutes les interprétations de variables I telles que $I(p) = V$ et avec toutes les interprétations de variables I telles que $I(p) = F$.

Séquents

Definition (Séquent)

Un *séquent* est un couple (P, c) composé d'un ensemble éventuellement vide P de formules logiques (les *prémisses*) et d'une formule c (la *conclusion*).

Les séquents sont la première étape dans la formalisation d'un raisonnement.



Véracité

Definition (Vrai)

Un séquent est *vrai dans une interprétation* I si, dans cette interprétation, les prémisses impliquent la conclusion.

Definition (Valide)

Un séquent est *valide* s'il est vrai dans toutes les interprétations.

Lorsqu'un séquent (P, c) est valide, on note $P \models c$.



À vous

Proposition

La formule f est vraie dans I si et seulement si le séquent (\emptyset, f) est vrai dans I .

Proposition

La formule f est valide si et seulement si le séquent (\emptyset, f) est valide.



Quelques règles

Proposition

$P \models c$ si et seulement si $P \models \text{NON NON } c$.

Proposition

Si $P \models \text{NON } (f \text{ IMPLIQUE } f')$ alors $P \models \text{NON } f'$ et $P \models f$.

Exercice

$P \cup \{f\} \models c$ si et seulement si $P \cup \{\text{NON NON } f\} \models \text{NON NON } c$.



Introduction

Booléens

Logique formelle

Logique propositionnelle

Interprétations

Séquents

Logiques

Termes de preuves

Preuves et programmes

Logique des prédicats

Conclusions



Inférence

Objectif Définir une logique.**Notation** Règle d'inférence (ou de déduction).

$$\text{Nom} \frac{\text{Prémises}}{\text{Conclusion}} \text{Conditions}$$

"Si les conditions sont remplies et si les prémisses peuvent être prouvées, alors, la règle Nom permet de prouver la conclusion."



Règles d'inférence

- ▶ Le nom est juste un nom.
- ▶ Les conditions peuvent être de n'importe quelle forme.
- ▶ Les prémisses sont de la forme $P_1 \vdash c_1, P_2 \vdash c_2 \dots$ c'est-à-dire "sous les hypothèses P_1 , on peut prouver c_1 , sous les hypothèses P_2 , on peut prouver $c_2 \dots$ ".
- ▶ La conclusion est de la forme $P \vdash c$, c'est-à-dire "sous l'hypothèse P , on peut prouver c ".



Exemple

$$\text{Premise} \frac{}{P \vdash f} f \in P$$

- ▶ Cette règle s'appelle *Premise*.
- ▶ La règle ne s'applique que si f est l'une des formules de P .
- ▶ Pas de prémisses = il n'y a rien à prouver. C'est un cas de base ou *axiome*.
- ▶ On cherche à prouver que $P \vdash f$, c'est-à-dire que sous les hypothèses P , la formule f est valide.



Exemple

$$\text{Premisse } \frac{}{P \vdash f} f \in P$$

"Pour prouver que f est vrai sous les hypothèses P , lorsque f est une des hypothèses, il suffit d'invoquer la règle Premisse."



Pourquoi ?

Note Ceci est uniquement une notation. Avant de s'en servir, il faut se débrouiller pour que les règles soient bonnes.

Definition (Logique)

Une *logique* \mathcal{L} est un ensemble de règles d'inférence.

Definition (Prouvable)

L'ensemble des propositions *prouvables* dans \mathcal{L} est l'ensemble construit par induction structurelle à partir de \mathcal{L} .



On continue ?

Q Tout le monde est d'accord ?

Note Sauf mention du contraire, tous les noms de variables, ensembles... qui apparaissent dans une règle d'inférence sont considérés comme quantifiés par \forall .



Logique des séquents

$$\begin{array}{ll} \text{Premisse } \frac{}{P \vdash f} f \in P & \text{Augmentation } \frac{P \vdash f}{P \cup \{a\} \vdash f} \\ \text{Modus Ponens } \frac{P \vdash f \quad P \vdash f \text{ IMPLIQUE } x}{P \vdash x} & \text{Synthèse } \frac{P \cup \{f\} \vdash x}{P \vdash f \text{ IMPLIQUE } x} \\ \text{Double négation 1 } \frac{P \vdash \text{NON NON } f}{P \vdash f} & \text{Double négation 2 } \frac{P \vdash f}{P \vdash \text{NON NON } f} \\ \text{Absurde } \frac{P \cup \{f\} \vdash x \quad P \cup \{f\} \vdash \text{NON } x}{P \vdash \text{NON } f} \end{array}$$



Vers la déduction...

Q Que sommes-nous en train de faire?

A Nous sommes en train de formaliser la notion de raisonnement.

A ... ce qui est une étape dans l'*automatisation* des mathématiques.



Preuves

Definition (Prouvable)

L'ensemble des propositions *prouvables* dans \mathcal{L} est l'ensemble construit par induction structurelle à partir de \mathcal{L} .

Definition (Preuve)

Une *preuve* est le détail de la construction d'un élément prouvable dans une logique.



Comment prouver

En logique formelle, prouver une affirmation, c'est

- ▶ écrire cette affirmation ;
- ▶ appliquer une règle dont la conclusion est cette affirmation ;
On est alors ramené à une ou plusieurs affirmations plus simples à prouver.
- ▶ appliquer à une de ces affirmations une règle dont la conclusion est cette affirmation ;
- ▶ ...
- ▶ jusqu'à avoir liquidé toutes les affirmations à prouver.



Exemple

Proposition

Pour toute formule f , nous avons $\emptyset \vdash f$ IMPLIQUE f .

Preuve

$$\text{?Synthèse} \frac{\text{?} \begin{array}{c} \text{?} \text{Prémisse} \\ \{f\} \vdash f \end{array}}{\emptyset \vdash f \text{ IMPLIQUE } f} \text{?}\emptyset$$



Encore un ?

Proposition

Pour toute formule f , nous avons $\emptyset \vdash f$ IMPLIQUE $(f$ IMPLIQUE $f)$.

$$\text{Synthèse} \frac{\text{Premisse} \frac{\emptyset}{\{f\} \cup \{f\} \vdash f} \quad \text{Synthèse} \frac{\{f\} \cup \{f\} \vdash f}{\{f\} \vdash f \text{ IMPLIQUE } f}}{\emptyset \vdash f \text{ IMPLIQUE } (f \text{ IMPLIQUE } f)}$$



On complique encore les choses

Proposition

Pour tout jeu de prémisses P et toutes formule f et g , nous avons $P \cup \{f\} \vdash g$ si et seulement si $P \cup \{\text{NON NON } f\} \vdash g$

Pour cette preuve, nous allons

- ▶ prouver le sens "seulement si"
 - ▶ supposer $P \cup \{\text{NON NON } f\} \vdash g$
 - ▶ prouver $P \cup \{f\} \vdash g$
- ▶ prouver le sens "si"
 - ▶ supposer $P \cup \{f\} \vdash g$
 - ▶ prouver $P \cup \{\text{NON NON } f\} \vdash g$



Seulement si

Supposons $P \cup \{\text{NON NON } f\} \vdash g$.

$$\text{MP} \frac{\text{DN2} \frac{\text{Premisse} \frac{\emptyset}{P \cup \{f\} \vdash f} \quad \text{Augm} \frac{\text{Synth} \frac{P \cup \{\text{NON NON } f\} \vdash g}{P \cup \{\text{NON NON } f\} \vdash \text{NON NON } f} \quad \text{DN2} \frac{P \cup \{f\} \vdash f}{P \cup \{f\} \vdash \text{NON NON } f}}{P \cup \{f\} \vdash \text{NON NON } f}}{P \cup \{f\} \vdash g}}{P \cup \{f\} \vdash g}$$



si

Supposons $P \cup \{f\} \vdash g$.

$$\text{MP} \frac{\text{DN1} \frac{\text{Premisse} \frac{\emptyset}{P \cup \{\text{NON NON } f\} \vdash \text{NON NON } f} \quad \text{Augm} \frac{\text{Synth} \frac{P \cup \{f\} \vdash g}{P \cup \{f\} \vdash \text{NON NON } f}}{P \cup \{\text{NON NON } f\} \vdash f}}{P \cup \{\text{NON NON } f\} \vdash f}}{P \cup \{\text{NON NON } f\} \vdash g}}{P \cup \{\text{NON NON } f\} \vdash g}$$



On continue à compliquer

Seulement si...

Proposition

Pour tout jeu de prémisses P et toutes formule f et g , nous avons $P \vdash f$ IMPLIQUE g si et seulement si $P \vdash (\text{NON } g)$ IMPLIQUE $(\text{NON } f)$

Pour cette preuve, nous allons

- ▶ prouver le sens "seulement si"
 - ▶ supposer $P \vdash f$ IMPLIQUE g
 - ▶ prouver $P \vdash (\text{NON } g)$ IMPLIQUE $(\text{NON } f)$
- ▶ prouver le sens "si"
 - ▶ supposer $P \vdash (\text{NON } g)$ IMPLIQUE $(\text{NON } f)$
 - ▶ prouver $P \vdash f$ IMPLIQUE g

Nous supposons $P \vdash f$ IMPLIQUE g

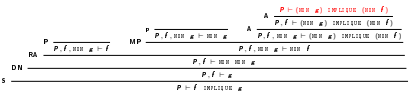


- A Augmentation
- RA Raisonnement par l'absurde
- P Prémisse
- S Synthèse
- DN Double négation
- MP Modus Ponens

...si

Bilan

Nous supposons $P \vdash (\text{NON } g)$ IMPLIQUE $(\text{NON } f)$



- Avec la logique des séquents, nous avons
- ▶ une notion de *prouvable*
 - ▶ des théorèmes sur des théorèmes
 - ▶ des preuves compréhensibles par un ordinateur.
- Il nous manque un lien entre *prouvable* et *valide*.

Croyez-moi sur parole

Théorème (Correction)

Dans la logique des propositions, tout séquent prouvable est valide.

Théorème (Complétude)

Dans la logique des propositions, tout séquent valide est prouvable.

Note Ceci est vrai uniquement parce que les règles d'induction ont été soigneusement choisies.

Ensemble des preuves

Q À quoi ressemble l'ensemble des preuves *en partant des hypothèses* Q ?

A C'est un ensemble défini par induction dont (en gros)

- ▶ la base est l'ensemble $Q \cup \{\epsilon\}$
- ▶ les opérateurs d'induction sont les noms des règles d'inférence employées.

En d'autres termes, nous avons un ensemble de preuves.

Termes de preuves

Cet ensemble de preuves peut être défini en OCaml :

```
type preuve = Premisse of formule          (**f**)
            | Augmentation of preuve = formule (**g*)
            | Modus Ponens of preuve        (**f*)
              = preuve                      (**f=>g*)
            | Synthèse of preuve
            | ...
```

Recherche de preuve

Q Comment trouver une preuve ?

A À la main.

A En essayant toutes les combinaisons possibles.

```
premise(P,F) <- in(F,P) .
augmentation(P,F) <- ajouter(P,G,P')...preuve(P',F) .
modusponens(P,F) <- preuve(P,implique(G,F)) . preuve(P,G) .
...
preuve(P,F) <- premise(P,F) .
preuve(P,F) <- augmentation(P,F) .
preuve(P,F) <- modusponens(P,F) .
```

Q Avantages et inconvénients de chaque approche ?

Pour information, Coq utilise une combinaison de ces deux approches.

Modus Ponens

Revenons au Modus Ponens

$$\text{Modus Ponens } \frac{P \vdash a : f \quad P \vdash b : f \text{ IMPLIQUE } \longrightarrow g}{P \vdash b : a : g}$$

Q Est-ce que ça vous rappelle quelque chose en OCaml?

A À peu de choses près, c'est la définition du typage d'une application de fonction en OCaml.

- ▶ P est la liste des variables, avec leur type
- ▶ a est une valeur
- ▶ b une fonction.



Synthèse

De la même manière

$$\text{Synthèse } \frac{P \cup \{f\} \vdash g}{P \vdash f \text{ IMPLIQUE } g} \quad \text{Synthèse } \frac{P \cup \{x : f\} \vdash e : g}{P \vdash a : (x : f) \longrightarrow (e : g)}$$

...

De même pour les autres règles.



Moralité

En programmation fonctionnelle, tout programme est une preuve.

Inversement, toute preuve est un programme.

Le système de types de OCaml n'est pas suffisamment puissant pour gérer des preuves non-triviales.

Celui de Coq l'est.

Ça en jette, hein ?



[Introduction](#)

[Booléens](#)

[Logique formelle](#)

[Logique propositionnelle](#)

[Interprétations](#)

[Séquents](#)

[Logiques](#)

[Termes de preuves](#)

[Preuves et programmes](#)

[Logique des prédicats](#)



[Conclusions](#)

Prédicats

La logique des prédicats étend la logique des propositions avec des variables non-booléennes, des ensembles et des quantificateurs.

Exemple IL EXISTE x DANS \mathbf{N} , *Premier*(x)

Exemple

POUR TOUT x DANS \mathbf{N} , (*Premier*(x) ET *Superieur*($x, 2$)) IMPLIQUE *Impair*(x)

Bien entendu, il est nécessaire de définir la signification de *Premier*, *Superieur*, *Impair*...

Grammaire

Cette grammaire emploie

- ▶ un ensemble de prédicats \mathcal{C} (ex: *Entier*, *Premier*, *Superieur*, *Impair*), qui généralise les variables booléennes
- ▶ un ensemble de nom de variables non-booléennes \mathcal{V} (ex: x , y).

$$\begin{array}{l} \mathcal{F} ::= \mathcal{N} \\ \quad | \text{ IL EXISTE } x, \mathcal{F} \quad x \in \mathbf{V}, e \in \mathcal{E} \\ \quad | \mathcal{F} \text{ IMPLIQUE } \mathcal{F} \\ \quad | \text{ NON } \mathcal{F} \\ \quad | r(x_1, \dots, x_n) \quad r \in \mathcal{C}, x_1 \dots x_n \in \mathbf{V} \end{array}$$

Note Dans $r(x_1, \dots, x_n)$, on peut avoir $n = 0$, c'est-à-dire une variable booléenne.

Deuxième couche

Comme précédemment, on définit ET et OU à partir de IMPLIQUE et NON .

De même, on définit POUR TOUT à partir de IL EXISTE (ou, au choix, IL EXISTE à partir de POUR TOUT), par "POUR TOUT x , f est un raccourci pour NON IL EXISTE x , NON f ".

Sémantique

Q De quoi avons-nous besoin pour donner un sens à une formule et ainsi déterminer si elle est vraie ?

A Il est nécessaire de généraliser la notion d'interprétation, pour prendre en compte les prédicats eux-mêmes.

Une a utre fois.

... et inférence

$$\text{Instantiation } \frac{P \vdash \text{POUR TOUT } x, f}{P \vdash f\{x \leftarrow y\}} \text{ y nouveau}$$

$$\text{Généralisation } \frac{P \vdash f}{P \vdash \text{POUR TOUT } x, f}$$

Plus les règles de la logique des propositions.

Arrêtons là

Il y a plein de théorèmes intéressants sur le sujet mais nous avons largement dépassé du programme.

Introduction

Booléens

Logique formelle

Logique propositionnelle

Interprétations

Séquents

Logiques

Termes de preuves

Preuves et programmes

Logique des prédicats

Conclusions

Bilan

Nous avons vu

- ▶ les booléens ;
- ▶ les bases de la logique formelle ;
- ▶ les bases de la preuve formelle ;
- ▶ quelques liens entre programmation fonctionnelle et la logique.

Ouvertures informatique

Ce que nous avons vu constitue

- ▶ les bases du système de types de OCaml
- ▶ les bases des termes de preuve de Coq
- ▶ les bases de Prolog
- ▶ les bases de la certification de programmes.



Ouvertures mathématiques

Ce que nous avons vu constitue

- ▶ les bases des logiques d'ordre supérieur
- ▶ les bases des logiques épistémiques
- ▶ les bases des logiques modales ;
- ▶ ...

