Logique et représentation des connaissances

Cours 1 - mercredi 17 septembre 2025

Nicolas Maudet (resp. AI2D)
Gauvain Bourgne (resp. MIND)
Djeser Kordon, Marie-Jeanne Lesot, Hénoïk Willot

LIP6, Sorbonne Université

Organisation générale

- Cours : Amphi B2, mercredi, 13h45-15h45
- TD/TME
  - G1 : lundi 13h45
  - G2 : mardi 8h30
  - G3 : jeudi 8h30
  - G4 : jeudi 8h30
- Site moodle : UE UM4IN800
  - https://moodle-sciences-25.sorbonne-universite.fr/
- Calendrier et salles
  - https://cal.ufr-info-p6.jussieu.fr/master/

Evaluation

- Barème
  - examen réparti 1 : 0.45
  - examen réparti 2 : 0.45
  - contrôle continu : 0.1
    - rendus de TME
    - mini-projet
- **Important :** la note de contrôle continu est conservée pour la note de seconde chance

Vue d'ensemble

Logique et représentation des connaissances

- Représentation de connaissances, cas général
  - logique des prédicats du premier ordre
  - logiques de description et ontologies
- Représentation de connaissances non factuelles
  - logiques modales, en particulier épistémique
  - cas de connaissances communes, partagées...
- Représentation de connaissances temporelles
  - intervalles d'Allen
  - réseaux de Petri
  - logiques temporelles

La logique formelle

- Objectifs :
  - modéliser le raisonnement
  - formaliser le raisonnement
  - automatiser le raisonnement
- Domaine “pas si ancien”
  - -500 – XVIIIème : Aristote, Socrate, Thomas d'Aquin, Leibniz, ...
  - XIXème : Boole, De Morgan, Peirce, Frege, ...
  - XXème : Hilbert, Russell, Herbrand, Skolem, Gentzen, Tarski, Gödel, Robinson, Kripke, ...
  - aujourd'hui : extensions, logiques non classiques, mises en œuvre et exploitations

Au programme du jour

- 1. Rappels de logique des prédicats du premier ordre, LPPO
  (notations LU2IN024)
  - 1. Syntaxe
  - 2. Sémantique
  - 3. Prouvabilité
- 2. Automatisation du raisonnement
  - 1. Méthode des tableaux sémantiques
  - 2. Méthode de résolution et prolog

Syntaxe de la logique du premier ordre

- **Eléments de base**
  - variables : X
  - constantes : $\mathcal{C} = \mathcal{F}_0$
  - fonctions : $\mathcal{F} = \bigcup_n \mathcal{F}_n$

  $\Longrightarrow$ **termes** : $\mathcal{T}(X, \mathcal{F})$
  définition inductive
  - $\mathcal{C} \cup X \subset \mathcal{T}(X, \mathcal{F})$
  - $f(t_1, \dots, t_n) \in \mathcal{T}(X, \mathcal{F})$
    si $f \in \mathcal{F}_n$ et $t_1, \dots, t_n \in \mathcal{T}(X, \mathcal{F})$

  - **prédicats** : $\mathcal{P} = \bigcup_n \mathcal{P}_n$

- **Formules logiques** $\mathcal{L}(X, \mathcal{F}, \mathcal{P})$: définition inductive
  - formule **atomique** : $F = P(t_1, \dots, t_n)$
    - où $t_1, \dots, t_n \in \mathcal{T}(X, \mathcal{F})$ et $P \in \mathcal{P}_n$
  - formule avec **connecteurs** : $\neg F, F \rightarrow G, F \wedge G, F \vee G$
    - si F, G sont des formules
  - formule avec **quantificateurs** : $\forall x F$ et $\exists x F$
    - si F est une formule et $x \in X$

Typologies

- Occurrence de variable
  - **liée** : sous la portée d'un quantificateur
  - **libre** sinon
- Variable
  - liée : **toutes** les occurrences sont liées
  - libre : **au moins une** occurrence libre
- Formule
  - **close** : toutes les variables sont liées

Au programme du jour

- 1. Rappels de logique des prédicats du premier ordre, LPPO
  - 1. Syntaxe
  - **2. Sémantique** (notations LU2IN024)
  - 3. Prouvabilité
- 2. Automatisation du raisonnement
  - 1. Méthode des tableaux sémantiques
  - 2. Méthode de résolution et prolog

Structure et valuation

- **Valeur de vérité dans $\mathbb{B}$** : dépend de la structure et de la valuation
- **Structure** $M = \langle |M|, \bullet^M \rangle$
  - $|M|$ domaine d'interprétation
  - chaque constante $k \in \mathcal{F}_0$ est associée à un élément $k^M \in |M|$
  - chaque symbole de fonction $f \in \mathcal{F}_n$ d'arité n est associé à une fonction n-aire $f^M : |M|^n \to |M|$
  - chaque prédicat $P \in \mathcal{P}_n$ d'arité n est associé à un ensemble de n-uplets $P^M \subseteq |M|^n$
- **Valuation** $v : X \to |M|$
  - fonction donnant une valeur à chaque variable

Formalisation du calcul du degré de vérité

- Etant donné une structure M et une valuation v
  - interprétation des termes : définition inductive

$$[t]_v^M = \begin{cases} v(x) & \text{si } t = x \in X \\ k^M & \text{si } t = k \in \mathcal{F}_0 \\ f^M([t_1]_v^M, \ldots, [t_n]_v^M) & \text{si } t = f(t_1, \ldots, t_n) \end{cases}$$

Formalisation du calcul du degré de vérité

- Etant donné une structure M et une valuation v
- $[F]_v^M$ valeur de vérité de F : définition inductive
  - pour chacun des trois types de formules
  - formule atomique $[P(t_1, \ldots, t_n)]_v^M = 1$ ssi $([t_1]_v^M, \ldots, [t_n]_v^M) \in P^M$
  - formule avec connecteur

$$[\neg F]_v^M = \overline{[F]_v^M} \qquad [F \wedge G]_v^M = [F]_v^M \cdot [G]_v^M$$
$$[F \vee G]_v^M = [F]_v^M + [G]_v^M \qquad [F \rightarrow G]_v^M = \overline{[F]_v^M} + [G]_v^M$$

  - formule avec quantificateur
    - $[\forall x F]_v^M = 1$ ssi $[F]_{v[x \leftarrow m]}^M = 1$ pour chaque $m \in |M|$
    - $[\exists x F]_v^M = 1$ ssi $[F]_{v[x \leftarrow m]}^M = 1$ pour au moins un $m \in |M|$

Validité et satisfiabilité

- Formule **universellement valide** : vraie quelles que soient M et v
  - $P(f(x)) \vee \neg P(f(x))$
- Formule **valide** : pour M donnée, vraie quelle que soit v
  - $P(a, y)$ pour $|M| = \mathbb{N}$, $a^M = 0$ et $P^M = \leq$
  - formule close vraie
- Formule **satisfiable** : pour M donnée, il existe v telle qu'elle soit vraie
  - $P(x, y)$ pour $|M| = \mathbb{N}$, $P^M = \leq$: considérer $v : x \mapsto 0, y \mapsto 1$
- Formule **insatisfiable** : pour M donnée, fausse quelle que soit v
  - rappel : F est insatisfiable ssi $\neg F$ est valide

Conséquence sémantique

- Conséquence sémantique : $\Delta \models F$
  - ssi quelles que soient M et v, si $[\Delta]_v^M = 1$ alors $[F]_v^M = 1$

- Exemples/exercices : avec $a \in \mathcal{F}_0$ et $p, q \in \mathcal{P}_0$ et $P, Q \in \mathcal{P}_1$
  - $q \models p \to q$
  - $p, p \to q \models q$
  - $P(a), \forall x(P(x) \to Q(x)) \models Q(a)$
  - $\exists x(P(x) \land Q(x)) \models \exists x P(x) \land \exists x Q(x)$
  - $\forall x(P(x) \lor Q(x)) \not\models \forall x P(x) \lor \forall x Q(x)$
  - $\exists x \forall y P(x, y) \models \forall y \exists x P(x, y)$

- Formule universellement valide : $\models F$

Théorème de la déduction

$$F \models G \qquad \Longleftrightarrow \qquad \models F \to G$$

- Preuve
  - application de la définition de $\models$ et de la sémantique de $\to$

Au programme du jour

- 1. Rappels de logique des prédicats du premier ordre, LPPO
  - 1. Syntaxe
  - 2. Sémantique
  - **3. Prouvabilité**
- 2. Automatisation du raisonnement
  - 1. Méthode des tableaux sémantiques
  - 2. Méthode de résolution et prolog

Prouvabilité

$$\Delta \vdash F$$

- Pas d'interprétation, de table de vérité, de vrai ou faux
- Un **système formel** est composé de
  - un langage formel
  - un ensemble de schémas d'axiome
  - un ensemble de règles d'inférence
- Quelques exemples
  - système de Hilbert
  - déduction naturelle : cf LU2IN024
  - calcul des séquents

Système de Hilbert propositionnel

- Langage formel
  - ensemble de variables propositionnelles
  - deux connecteurs : $\neg$ et $\rightarrow$
- Trois **schémas d'axiomes**
  - SA1 : $A \rightarrow (B \rightarrow A)$
  - SA2 : $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
  - SA3 : $(\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$
- Une règle d'inférence : le **modus ponens**
  - MP : $A, A \rightarrow B \vdash B$

Système de Hilbert propositionnel

- **Déduction de A dans une théorie Δ** : $\Delta \vdash A$
  suite finie de formules $F_0, \ldots, F_p$ telle que $F_p = A$ et pour tout i
  - F_i est l'instanciation de l'un des axiomes
  - ou F_i est l'une des hypothèses : $F_i \in \Delta$
  - ou F_i est obtenue par MP appliqué à F_j et F_k avec $j < i$ et $k < i$

- Exemple : montrer que $\vdash F \to F$

Théorème de la déduction

$$A_1, \ldots, A_{n-1}, A_n \vdash B \quad \Longleftrightarrow \quad A_1, \ldots, A_{n-1} \vdash A_n \rightarrow B$$

- Preuve de $\Leftarrow$
  - $i = 1..n$: $F_i = A_i$
  - $F_{n+1} = A_n \rightarrow B$ par hypothèse et en utilisant F_i, $i = 1..n-1$
  - $F_{n+2} = B$ MP sur F_n et F_{n+1}
- Preuve de $\Rightarrow$: 4 cas à considérer
  - B est un axiome
  - B est l'un des A_i, $i = 1..n-1$
  - $B = A_n$
  - B est obtenu par application du MP à $B' \rightarrow B$ et B'
    preuve par induction sur la taille de la démonstration

Complétude et adéquation

$$\Delta \models F \quad \Longleftrightarrow \quad \Delta \vdash F$$

Au programme du jour

- 1. Rappels de logique des prédicats du premier ordre, LPPO
  - 1. Syntaxe
  - 2. Sémantique
  - 3. Prouvabilité
- **2. Automatisation du raisonnement**
  - 1. Méthode des tableaux sémantiques
  - 2. Méthode de résolution et prolog $\longrightarrow$ cours 4

Objectifs et méthodes

- Démonstration automatique
  - **algorithmes et implémentations !**

- Problèmes des méthodes précédentes
  - tables de vérité : logique des propositions + complexité exponentielle
  - méthodes formelles : non automatisables

- Approches présentées en LRC
  - méthode des tableaux sémantiques :
    - logiques propositionnelle, de description, modales
  - résolution :
    - cas propositionnel, unification, cas premier ordre : prolog

Principe général de la méthode des tableaux

- Entrée : Δ
  - un ensemble de formules
  - interprétation conjonctive
- Méthode sémantique
  - établir si Δ est satisfiable ou insatisfiable
- Approche syntaxique
  - décomposition itérative de formules en sous-formules
  - construction d'un arbre
  - analyse des feuilles

Algorithme

- Initialisation
  - construire un nœud racine, étiqueté par l'ensemble Δ
  - le marquer comme non traité
- Itérativement
  - choisir un nœud non traité et le marquer comme traité
  - s'il contient 2 littéraux complémentaires : **fermé** ou clash
  - sinon, s'il contient uniquement des formules atomiques : **ouvert**
  - sinon, appliquer les règles de réécriture : cf transparent suivant
- Quand il n'y a plus de nœud non traité
  - si l'arbre contient **une feuille ouverte : Δ satisfiable**
  - si **toutes les feuilles sont fermées : Δ insatisfiable**

Etape de réécriture

- Dans un nœud étiqueté par un ensemble Δ de formules
- Choisir une formule $F \in \Delta$ non atomique
- Si elle est de type α
  - créer un sous-nœud marqué comme non traité
  - lui associer l'étiquette $\Delta \setminus \{F\} \cup \{\alpha_1, \alpha_2\}$
  - où α_1 et α_2 sont les formules obtenues par réécriture de F
- Si elle est de type β
  - créer deux sous-nœuds marqués comme non traités
  - associer au premier l'étiquette $\Delta \setminus \{F\} \cup \{\beta_1\}$
  - associer au second l'étiquette $\Delta \setminus \{F\} \cup \{\beta_2\}$
  - où β_1 et β_2 sont les formules obtenues par réécriture de F

Règles de réécriture

- Formules de type α

F	α_1	α_2
$\neg\neg\varphi$	φ	
$\varphi_1 \wedge \varphi_2$	φ_1	φ_2
$\neg(\varphi_1 \vee \varphi_2)$	$\neg\varphi_1$	$\neg\varphi_2$
$\neg(\varphi_1 \rightarrow \varphi_2)$	φ_1	$\neg\varphi_2$
$\varphi_1 \leftrightarrow \varphi_2$	$\varphi_1 \rightarrow \varphi_2$	$\varphi_2 \rightarrow \varphi_1$

- Formules de type β

F	β_1	β_2
$\varphi_1 \vee \varphi_2$	φ_1	φ_2
$\neg(\varphi_1 \wedge \varphi_2)$	$\neg\varphi_1$	$\neg\varphi_2$
$\varphi_1 \rightarrow \varphi_2$	$\neg\varphi_1$	φ_2
$\neg(\varphi_1 \leftrightarrow \varphi_2)$	$\neg(\varphi_1 \rightarrow \varphi_2)$	$\neg(\varphi_2 \rightarrow \varphi_1)$

Intuition

- Pour que Δ soit satisfiable
  - il faut nécessairement que F soit satisfaite
- Pour que F soit satisfaite
  - il faut que α_1 **et** α_2 soient satisfaites
    $\Longrightarrow$ un seul sous-nœud où F est remplacée par α_1 et α_2
  - il faut que β_1 **ou** β_2 soient satisfaites
    $\Longrightarrow$ deux sous-branches à explorer,
    où F est remplacée par β_1 et β_2 respectivement

Interprétation finale de l'arbre construit

- S'il y a **une feuille ouverte** : c'est un modèle (partiel)
  $\Longrightarrow$ **Δ est satisfiable**

- **Si toutes les feuilles sont fermées** : il n'y a pas de modèle
  $\Longrightarrow$ **Δ est insatisfiable**

- Si **toutes les feuilles sont ouvertes**
  $\Longrightarrow$ **ATTENTION !! on ne conclut rien de spécial !**
  - juste que la formule est satisfiable
  - voir par exemple $\Delta = \{p \vee q\}$
  - il faudrait examiner si les feuilles couvrent tous les cas possibles

- Question additionnelle essentielle
  - comment prouver qu'une formule est valide ?

Propriétés

- Méthode de réfutation, initialement
  - pour prouver l'insatisfiabilité d'un ensemble de formules
  - extension dans le cas d'une feuille ouverte

- Les règles fournies sont telles que
  la méthode des tableaux sémantiques
  est adéquate et complète

- Stratégies d'implémentation :
  - heuristiques de choix des nœuds et des formules traités
  - typiquement, formules α prioritaires