# UE Modélisation et Programmation

## Matière Modélisation

- Responsable UE et matière : Marc Pantel
  `http://pantel.perso.enseeiht.fr`
- Cours : 5 séances (Marc Pantel)
- Travaux Dirigés : 7 séances
- Travaux Pratiques en binôme : 7 séances
- Examen écrit 60% : Similaire TD, Formulaire fourni (sans autres documents)
- Bureau d'étude en temps limité 40% : Similaire TP
- Attention : Le vocabulaire et les notations dans ce domaine sont multiples et pas encore stabilisés. Cette matière est la synthèse de nombreux documents. Il n'y a pas d'ouvrage de référence associé.

# Objectifs

## Informatique et Mathématiques

- Les sciences physiques, de la vie, humaines et sociales s'appuient sur des modèles mathématiques pour la compréhension, la prédiction et la prescription (définition d'un nouveau produit).
  - Modèle en science (compréhension et prédiction) : Approximation de la réalité
    Le modèle doit être aussi proche que possible de la réalité.
  - Modèle en ingénierie : Prescription de la réalité
    La réalité doit être aussi proche que possible du modèle.
- Qu'en est il de l'informatique ?
  - Science formelle similaire aux mathématiques
  - Thèse de Church-Turing (calculabilité) :
    Les programmes informatiques permettent de calculer les même fonctions que les mathématiques
  - Correspondance/Isomorphisme de Curry-Howard :
    Un programme $P$ bien typé de type $\tau$ est isomorphe à la preuve en logique constructive d'une formule isomorphe à $\tau$
- Etude des techniques de modélisation de programmes et de langages
- Etude des techniques de preuve de programmes

# Plan

## Modélisation

- Modélisation et Preuve de programmes
  - Logique des propositions : C1, TD1, TD2, TP1
  - Logique des prédicats : C2, TD3, TP2
  - Preuve de programmes fonctionnels : C3, TD4, TP3
  - Preuve de programmes impératifs : C4, TD5, TP4
- Modélisation des langages : C5, TD6, TD7, TP5, TP6, TP7
  - Théorie des langages
  - Expressions régulières, Grammaires
  - XML, JSON

# Notation

## Règles de déduction

- Soient $J_1, \ldots J_n$ et $J$ des jugements :

| | Notation | Signification |
|---|---|---|
| Déduction | $\dfrac{J_1 \quad J_n}{J}$ | si $J_1$ et $\ldots$ et $J_n$ sont valides alors $J$ est valide |
| Axiome | $\dfrac{}{J}$ | $J$ est valide |

- sémantique : $(\bigwedge_{i \in [1 \cdots n]} J_i) \rightarrow J$ et $\top \rightarrow J$
- méthode de chaînage arrière :
  pour prouver $J$, il suffit de prouver $J_1$ et $\ldots$ et $J_n$
- Exemples de jugements :
  - Typage : $x_1 : \tau_1, \ldots, x_n : \tau_n \vdash e : \tau$
  - Calcul : $x_1 \mapsto v_1, \ldots, x_n \mapsto v_n \vdash e \Rightarrow v$
  - Preuve : $H_1, \ldots, H_n \vdash \varphi$

# Systèmes formels

## Définitions

- Syntaxe concrète : Vision utilisateur
  Logique : Formule avec constantes, variables, opérateurs, lieurs (définition variables) et parenthèses
- Syntaxe abstraite : Vision information structurée
  Logique : Arbre étiqueté par constantes, variables, opérateurs, lieurs sans parenthèses
- Sémantique : Signification
  Logique : Valide, Satisfiable, Invalide, Insatisfiable, Inconnue
  Notation $\models \varphi$
- Axiomatisation de la sémantique
  Modélise la sémantique par la construction de preuves (démonstration)
  Approche syntaxique de la sémantique
  Logique : Notation $\vdash \varphi$
- Mécanisation de l'axiomatisation
  Construction automatique des preuves

# Systèmes formels

## Propriétés souhaitées

- Consistance sémantique : La sémantique ne peut pas être
  - Valide et Invalide
  - Satisfiable et Insatisfiable
- Complétude sémantique : La sémantique est toujours
  - Valide ou Invalide
  - Satisfiable ou Insatisfiable
  - Jamais inconnue
- Correction axiomatisation : $\forall \varphi.\ \vdash \varphi \rightarrow \models \varphi$
- Complétude axiomatisation : $\forall \varphi.\ \models \varphi \rightarrow \vdash \varphi$
  Exemple : Incomplétude de l'arithmétique (théorème de Gödel)
- Décidabilité : Mécanisation construit une preuve en temps fini
- Semi-décidabilité : Mécanisation calcule en temps fini quand la preuve existe (valide, satisfiable)
- Indécidabilité : Mécanisation peut ne pas se terminer
  Exemple : Test d'arrêt de la machine de Turing

# Syntaxe

## Vision algébrique

- Notons $\Phi$ l'ensemble dénombrable des formules bien formées de logique des propositions
- Eléments lexicaux :
  - Propositions (variables propositionnelles) : mots, phrases, ... (ensemble $\mathcal{P}$ dénombrables)
  - Opérateurs : $\bot, \top, \neg, \vee, \wedge, \rightarrow, \leftrightarrow$
  - Contrôle structure (associativité, priorité) : $(,)$
- Eléments grammaticaux :
  - Constantes (Opérateurs zéro-aire) : Propositions, $\top$ (Té) et $\bot$ (Anti-Té)
  - Opérateur unaire : $\neg$ (Négation)
  - Opérateurs binaires associatifs et commutatifs : $\vee$ (disjonction), $\wedge$ (conjonction)
  - Opérateur binaire commutatif : $\leftrightarrow$ (équivalence)
  - Opérateur binaire associatif à droite : $\rightarrow$ (implication)
  - Priorité croissante : $\rightarrow, \leftrightarrow, \vee, \wedge, \neg$

# Syntaxe

## Vision déductive

Soit $\mathcal{P}$ un ensemble dénombrable de variables propositionnelles

- Version classique

  Axiomes $\quad \dfrac{}{\top \in \Phi} \quad \dfrac{}{\bot \in \Phi} \quad \dfrac{}{P \in \Phi}\ (P \in \mathcal{P})$

  Déductions $\quad \dfrac{\varphi \in \Phi}{\neg\varphi \in \Phi} \quad \dfrac{\varphi \in \Phi \qquad \psi \in \Phi}{(\varphi\ op\ \psi) \in \Phi}\ (op \in \{\wedge, \vee, \rightarrow, \leftrightarrow\})$

- Version stratifiée (élimination paradoxe de Russel)

  $$\Phi = \bigcup_{i \in \mathbb{N}} \Phi_i$$

  $$\frac{}{\top \in \Phi_0} \quad \frac{}{\bot \in \Phi_0} \quad \frac{}{P \in \Phi_0}\ (P \in \mathcal{P})$$

  $$\frac{\varphi \in \Phi_n}{\neg\varphi \in \Phi_{n+1}} \quad \frac{\varphi \in \Phi_m \qquad \psi \in \Phi_n}{(\varphi\ op\ \psi) \in \Phi_{m+n+1}}\ (op \in \{\wedge, \vee, \rightarrow, \leftrightarrow\})$$

# Sémantique

## Tables de vérité

- Valeurs de vérité notées $V$ (vrai) et $F$ (faux)
  Autres notations possibles ($T$ et $F$, 1 et 0, ...)
- Opérateurs définis pour chaque valeur de vérité des opérandes

| $\neg$ | |
|---|---|
| $F$ | $V$ |
| $V$ | $F$ |

| $\wedge$ | $F$ | $V$ |
|---|---|---|
| $F$ | $F$ | $F$ |
| $V$ | $F$ | $V$ |

| $\vee$ | $F$ | $V$ |
|---|---|---|
| $F$ | $F$ | $V$ |
| $V$ | $V$ | $V$ |

| $\rightarrow$ | $F$ | $V$ |
|---|---|---|
| $F$ | $V$ | $V$ |
| $V$ | $F$ | $V$ |

| $\leftrightarrow$ | $F$ | $V$ |
|---|---|---|
| $F$ | $V$ | $F$ |
| $V$ | $F$ | $V$ |

- Notation sous la forme de formules élémentaires :

| $A$ | $B$ | $\neg A$ | $A \wedge B$ | $A \vee B$ | $A \rightarrow B$ | $A \leftrightarrow B$ |
|---|---|---|---|---|---|---|
| $F$ | $F$ | $V$ | $F$ | $F$ | $V$ | $V$ |
| $F$ | $V$ | $V$ | $F$ | $V$ | $V$ | $F$ |
| $V$ | $F$ | $F$ | $F$ | $V$ | $F$ | $F$ |
| $V$ | $V$ | $F$ | $V$ | $V$ | $V$ | $V$ |

# Sémantique

## Construction tables de vérité

- Formule $\varphi \in \Phi$ contient variables $\{P_i\}_{i \in [1 \cdots n]} \subseteq \mathcal{P}$
- Variables propositionnelles $P_i$ reçoivent valeurs de vérité
- $n$ variables : $2^n$ lignes
- Discriminant ligne $\bigwedge_{i \in [1 \cdots n]} \alpha_i$ avec : $\begin{cases} \alpha_i = P_i \text{ si valeur } V \\ \alpha_i = \neg P_i \text{ si valeur } F \end{cases}$
- 1 colonne par variable propositionnelle
- 1 colonne par opérateur de la formule
- dont 1 colonne pour la formule complète

# Sémantique

## Exemple de tables de vérité

- Formule : $(A \wedge B) \rightarrow (B \vee A)$
- Table de vérité

| Discriminant | $A$ | $B$ | $A \wedge B$ | $B \vee A$ | $(A \wedge B) \rightarrow (B \vee A)$ |
|---|---|---|---|---|---|
| $\neg A \wedge \neg B$ | $F$ | $F$ | $F$ | $F$ | $V$ |
| $\neg A \wedge B$ | $F$ | $V$ | $F$ | $V$ | $V$ |
| $A \wedge \neg B$ | $V$ | $F$ | $F$ | $V$ | $V$ |
| $A \wedge B$ | $V$ | $V$ | $V$ | $V$ | $V$ |

# Sémantique

## Vocabulaire

Selon sa table de vérité, $\varphi \in \Phi$ est :

- Valide, tautologie, ... :
  Toutes les lignes $V$
  Notée $\models \varphi$
- Satisfiable, consistante, cohérente, ... :
  Au moins une ligne $V$ (modèle de $\varphi$)
  Si $L$ est son discriminant alors $\models L \rightarrow \varphi$
  Notée $\neg \models \neg\varphi$
  Si Valide alors Satisfiable
- Invalide, ... :
  Au moins une ligne $F$
  Si et seulement si $\neg\varphi$ satisfiable
  Notée $\neg \models \varphi$
- Insatisfiable, inconsistante, incohérente, antilogie, ... :
  Toutes les lignes $F$
  Si et seulement si $\neg\varphi$ valide
  Notée $\models \neg\varphi$
  Si Insatisfiable alors Invalide

# Sémantique

## Relation d'équivalence

- Soient $\varphi, \psi, \chi \in \Phi$ :
- $\varphi = \psi$ si et seulement si $\varphi$ et $\psi$ ont la même table de vérité
- $\varphi = \psi$ si et seulement si $\models \varphi \leftrightarrow \psi$
- Equivalence de $\rightarrow$ et $\leftrightarrow$ :

$$\varphi \rightarrow \psi = \neg\varphi \vee \psi$$
$$\varphi \leftrightarrow \psi = (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$$
$$\varphi \leftrightarrow \psi = (\varphi \wedge \psi) \vee (\neg\varphi \wedge \neg\psi)$$

- Lois de De Morgan :

$$\neg(\varphi \wedge \psi) = \neg\varphi \vee \neg\psi$$
$$\neg(\varphi \vee \psi) = \neg\varphi \wedge \neg\psi$$

- Opposé, éléments neutres et absorbants :

$\varphi \wedge \neg\varphi = \bot$ $\quad$ $\varphi \rightarrow \varphi = \top$ $\quad$ $\varphi \wedge \bot = \bot$ $\quad$ $\varphi \vee \bot = \varphi$ $\quad$ $\neg\neg\varphi = \varphi$

$\varphi \vee \neg\varphi = \top$ $\quad$ $\varphi \leftrightarrow \varphi = \top$ $\quad$ $\varphi \wedge \top = \varphi$ $\quad$ $\varphi \vee \top = \top$

# Sémantique

## Relation d'équivalence

- Soient $\varphi, \psi, \chi \in \Phi$ :
- Idempotence :

$$\varphi \wedge \varphi = \varphi \quad \varphi \vee \varphi = \varphi$$

- Commutativité :

$$\varphi \wedge \psi = \psi \wedge \varphi \quad \varphi \vee \psi = \psi \vee \varphi \quad \varphi \leftrightarrow \psi = \psi \leftrightarrow \varphi$$

- Associativité :

$$\begin{array}{ccccc} (\varphi \wedge \psi) \wedge \chi & = & \varphi \wedge \psi \wedge \chi & = & \varphi \wedge (\psi \wedge \chi) \\ (\varphi \vee \psi) \vee \chi & = & \varphi \vee \psi \vee \chi & = & \varphi \vee (\psi \vee \chi) \\ (\varphi \rightarrow \psi) \rightarrow \chi & \neq & \varphi \rightarrow \psi \rightarrow \chi & = & \varphi \rightarrow (\psi \rightarrow \chi) \end{array}$$

- Distributivité :

$$\varphi \wedge (\psi \vee \chi) = (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$$
$$\varphi \vee (\psi \wedge \chi) = (\varphi \vee \psi) \wedge (\varphi \vee \chi)$$

- Simplification :

$$\varphi \vee (\neg\varphi \wedge \psi) = \varphi \vee \psi \qquad \varphi \vee (\varphi \wedge \psi) = \varphi$$
$$\varphi \wedge (\neg\varphi \vee \psi) = \varphi \wedge \psi \qquad \varphi \wedge (\varphi \vee \psi) = \varphi$$

# Sémantique

## Formes normales

Pour toute formule $\varphi \in \Phi$, il existe :

- Une formule équivalente en forme normale disjonctive :

$$\begin{aligned}\varphi &= \bigvee_{i\in[1\cdots n]} \beta_i \\ \beta_i &= \bigwedge_{j\in[1\cdots m_i]} \alpha_{i,j} \\ \alpha_{i,j} &\in \mathcal{P} \cup \{\neg P \mid P \in \mathcal{P}\}\end{aligned}$$

- Une formule équivalente en forme normale conjonctive :

$$\begin{aligned}\varphi &= \bigwedge_{i\in[1\cdots n]} \beta_i \\ \beta_i &= \bigvee_{j\in[1\cdots m_i]} \alpha_{i,j} \\ \alpha_{i,j} &\in \mathcal{P} \cup \{\neg P \mid P \in \mathcal{P}\}\end{aligned}$$

- Ces formules sont obtenues en :
  - Remplaçant $\rightarrow$ et $\leftrightarrow$ par leurs équivalents
  - Rapprochant les négations $\neg$ des variables propositionnelles
  - Effectuant les distributivités de $\wedge$ sur $\vee$ (respectivement de $\vee$ sur $\wedge$)

# Sémantique

## Exemple d'équivalence sémantique

- Formule : $(A \to B) \leftrightarrow (\neg B \to \neg A)$
- Raisonnement équationnel
  - Remplacer $\to$ et $\leftrightarrow$ par leurs équivalents
    $((\neg A \vee B) \wedge (\neg\neg B \vee \neg A)) \vee (\neg(\neg A \vee B) \wedge \neg(\neg\neg B \vee \neg A))$
  - Rapprocher les négations $\neg$ des variables propositionnelles
    $((\neg A \vee B) \wedge (B \vee \neg A)) \vee (\neg(\neg A \vee B) \wedge \neg(B \vee \neg A))$
    $((\neg A \vee B) \wedge (B \vee \neg A)) \vee ((\neg\neg A \wedge \neg B) \wedge (\neg B \wedge \neg\neg A))$
    $((\neg A \vee B) \wedge (B \vee \neg A)) \vee ((A \wedge \neg B) \wedge (\neg B \wedge A))$
  - Simplification par Idempotence et Commutativité
    $((\neg A \vee B) \wedge (B \vee \neg A)) \vee (A \wedge \neg B)$
  - Distributivité
    $((\neg A \wedge (B \vee \neg A)) \vee (B \wedge (B \vee \neg A))) \vee (A \wedge \neg B)$
    $(((\neg A \wedge B) \vee (\neg A \wedge \neg A)) \vee ((B \wedge B) \vee (B \wedge \neg A))) \vee (A \wedge \neg B)$
  - Simplification par Associativité, Idempotence et Commutativité
    $((\neg A \wedge B) \vee \neg A) \vee (B \vee (A \wedge \neg B))$
  - Simplification
    $\neg A \vee (B \vee A)$
    $\top$

# Sémantique

## Base minimale d'opérateurs

- La mise en forme normale montre que $\{\vee, \wedge, \neg\}$ sont suffisants pour représenter toute formule
- Il existe des bases minimales d'opérateurs
  - $\{\wedge, \neg\}$ ou $\{\vee, \neg\}$ par De Morgan
  - $\{\rightarrow, \neg\}$ car $\varphi \vee \psi = \neg\varphi \rightarrow \psi$
  - $\{\rightarrow, \bot\}$ car $\neg\varphi = \varphi \rightarrow \bot$

# Déduction naturelle

## Cadre général

- Axiomatisation par des règles de déduction
- Approche par chaînage arrière : De la conclusion aux hypothèses
- Jugement $\Gamma \vdash \psi$ avec $\Gamma = \varphi_1, \ldots, \varphi_n$
  et $\varphi_1, \ldots, \varphi_n, \psi \in \Phi$
  $\varphi_i$ sont les hypothèses disponibles pour prouver $\psi$
- Sémantique : $\bigwedge_{i \in [1 \cdots n]} \varphi_i \rightarrow \psi$
- Axiome de l'hypothèse :
  $$\frac{}{\Gamma, \varphi \vdash \varphi}\ \text{Hyp}$$

# Déduction naturelle

## Règles de déduction constructive

| Introduction | Élimination |
|---|---|
| $\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ I_\to$ | $\dfrac{\Gamma \vdash \varphi \to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}\ E_\to$ |
| $\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}\ I_\wedge$ | $\dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi}\ E_\wedge^G \quad \dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}\ E_\wedge^D$ |
| $\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}\ I_\vee^G \quad \dfrac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}\ I_\vee^D$ | $\dfrac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \chi \quad \Gamma, \psi \vdash \chi}{\Gamma \vdash \chi}\ E_\vee$ |
| $\dfrac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi}\ I_\neg$ | $\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot}\ E_\neg$ |
| $\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot}\ I_\bot$ | $\dfrac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}\ E_\bot$ |

# Déduction naturelle

## Heuristique/Méthode de preuve

- Construire la preuve de bas en haut en appliquant par ordre de préférence :
- Les axiomes (règle de l'hypothèse, . . . ) ;
- Les régles d'élimination sur les hypothèses pour extraire la conclusion si elle figure dans une hypothèse ;
- Les règles d'introduction pour décomposer la conclusion jusqu'à obtenir un élément disponible dans les hypothèses ou une variable ;
- La règle $E_\bot$ (preuve par l'absurde constructive) s'il n'est pas possible de faire apparaitre en conclusion un élément figurant dans les hypothèses.
- Exemple

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \varphi \wedge \psi \vdash \varphi \wedge \psi}\,\text{Hyp}}{\Gamma, \varphi \wedge \psi \vdash \psi}\,E^G_\wedge \qquad \dfrac{\dfrac{}{\Gamma, \varphi \wedge \psi \vdash \varphi \wedge \psi}\,\text{Hyp}}{\Gamma, \varphi \wedge \psi \vdash \varphi}\,E^D_\wedge}{\Gamma, \varphi \wedge \psi \vdash \psi \wedge \varphi}\,I_\wedge}{\Gamma \vdash (\varphi \wedge \psi) \to (\psi \wedge \varphi)}\,I_\to$$

# Déduction naturelle

## Logique constructive et classique

- Approche philosophique
- Interdiction du tiers-exclus (Axiome $\varphi \vee \neg\varphi$)
- Interdiction de l'axiome du choix
- La logique classique consiste à utiliser les règles :

| Tiers-exclu | Preuve par l'absurde |
|---|---|
| $\dfrac{}{\Gamma \vdash \varphi \vee \neg\varphi}$ TE | $\dfrac{\Gamma, \neg\varphi \vdash \bot}{\Gamma \vdash \varphi}$ Abs |

- La preuve par l'absurde exploite le tiers-exclu

$$\dfrac{\dfrac{}{\Gamma \vdash \varphi \vee \neg\varphi}\ \text{TE} \qquad \dfrac{}{\Gamma, \varphi \vdash \varphi}\ \text{Hyp} \qquad \dfrac{\Gamma, \neg\varphi \vdash \bot}{\Gamma, \neg\varphi \vdash \varphi}\ E_\bot}{\Gamma \vdash \varphi}\ E_\vee$$

# Logique des propositions

## Conclusion

La logique des propositions est :

- Complète sémantiquement et axiomatiquement
- Consistante sémantiquement
- Correcte axiomatiquement
- Décidable mécaniquement
- Mais Très peu expressive
- Introduction des quantificateurs, des relations et des structures : Logique des prédicats

# Logique des propositions

## Mise en pratique

### L'assistant de preuve Coq

- Développé au sein d'INRIA
- Système $F$ : $\lambda$-calcul typé second ordre (Girard et Reynolds)
- Calcul des constructions inductives (Coquant)
- Correspondance de Curry-Howard
  - Formule = Type
  - Preuve = Programme

### Le langage de développement prouvé Why3

- Développé au sein du LRI et d'INRIA
- Logique des prédicats du premier ordre et Logique de Hoare
- Passerelle vers de nombreux outils de vérification :
  - Automatique : SAT solver (résolution par saturation), SMT (SAT Modulo Theory)
  - Semi-automatique : Assistants de preuve