Sémantique de la logique

- Comment écrire les formules ?
  - Aspects syntaxiques
- Comment déterminer la valeur de vérité d'une formule ?
  - Aspects sémantiques
  - logique bivaluée : vrai, faux
  - interprétation
- Comment démontrer de nouveaux résultats ?
  - Aspects déductifs
  - Conséquence logique
  - démonstration
  - règles de déduction

Interprétation (1/2)

- <u>*But*</u> : donner une valeur de vérité aux formules
- Une *interprétation* I d'une formule F est basée sur un ensemble de définition D, non vide, appelé domaine
  - à chaque symbole de constante de F est associé un élément de D
  - à chaque symbole de variable de F est associé la variable elle-même
  - à chaque symbole de fonction de F est associée une fonction de D^n dans D
  - à chaque symbole de prédicat de F est associé une fonction de D^n dans {0,1}
  - à chaque connecteur d'arité i est associée une fonction de $\{0,1\}^i$ dans {0,1}

a	¬a
1	0
0	1

∨	1	0
1	1	1
0	1	0

∧	1	0
1	1	0
0	0	0

⇒	1	0
1	1	0
0	1	1

Interprétation (2/2)

- si $F = \forall x\ G(x,y_1,..,y_n)$ (G formule dépendant de x et des variables libres $y_1,..,y_n$), pour tout $(a_1,..,a_n)$ de D^n, $I(F)(a_1,..,a_n)$ vaut 1 si pour tout a de D, $I(G)(a, a_1,..,a_n) = 1$, et vaut 0 sinon
- si $F = \exists\ x\ G(x,y_1,..,y_n)$ (G formule dépendant de x et des variables libres $y_1,..,y_n$), pour tout $(a_1,..,a_n)$ de D^n, $I(F)(a_1,..,a_n)$ vaut 1 s'il existe a de D telle que $I(G)(a, a_1,..,a_n) = 1$, et vaut 0 sinon

- Toute formule close peut donc être interprétée dans {0,1}
- Une interprétation d'une formule contenant i variables libres donne une application de D^i dans {0,1}. Une formule peut être ainsi vue comme une fonction booléenne de ses variables libres dans {0,1}.
- Une interprétation d'une formule est un modèle de cette formule si la formule est vraie pour cette interprétation

Exemple d'interprétation

- Interprétons $F = \forall x\ p(x) \Rightarrow q(x)$ sur le domaine {a,b,c}
- Une interprétation possible de la formule

x	I(p)(x)	I(q)(x)	$I(p \Rightarrow q)(x)$
a	1	1	1
b	0	0	1
c	1	0	1

=> F est vraie

- Autre exemple d'interprétation

x	I(p)(x)	I(q)(x)	$I(p \Rightarrow q)(x)$
a	0	1	0
b	0	1	0
c	1	0	1

=> F est fausse

Validité

- Une formule est valide (tautologie) si elle est vraie quelque soit l'interprétation (si toute interprétation est un modèle)
  - *exemple : $\forall x \neg p(x) \vee p(x)$ est une tautologie*

- Une formule est consistante (ou satisfiable) s'il existe une interprétation dans laquelle elle est vraie
  - *exemple : $\exists x \neg p(x)$*

- Une formule est insatisfiable (ou inconsistante) s'il n'existe pas d'interprétation dans laquelle elle est vraie
  - *exemple : $\neg p(x) \wedge p(x)$*

- *Note : une formule peut être invalide et consistante*

Equivalence

- Deux formules *f* et *f'* sont sémantiquement équivalentes si pour toute interprétation *I*, $I(f) = I(f')$, c'est à dire que leur tables de vérité sont les mêmes (on note $f \equiv f'$)

- *Quelques équivalences utiles* :
  - $p \Rightarrow q \equiv \neg p \vee q$
  - $p \wedge \neg p \equiv 0$
  - $p \vee \neg p \equiv 1$
  - $\neg (\neg p) \equiv p$
  - $\neg (p \wedge q) \equiv \neg p \vee \neg q$ (loi de Morgan)
  - $\neg (p \vee q) \equiv \neg p \wedge \neg q$ (loi de Morgan)
  - $p \wedge q \equiv q \wedge p$ et $p \vee q \equiv q \vee p$ (symétries de $\wedge$ et $\vee$)
  - associativités de $\wedge$ et $\vee$
  - $p \vee 1 \equiv 1$ et $p \wedge 0 \equiv 0$ (absorption)
  - $p \wedge 1 \equiv p$ et $p \vee 0 \equiv p$ (élément neutre)

Satisfaction

- Un ensemble de formules $\{f_1, .., f_n\}$ satisfait une formule f si pour toute interprétation I, pour tout i=1..n, si $I(f_i) = 1$ alors $I(f) = 1$, c'est-à-dire si tout modèle de $\{f_1, .., f_n\}$ est aussi modèle de f. On note $\{f_1, .., f_n\} \models f$. On dit aussi que f est conséquence logique de $\{f_1, .., f_n\}$

- Si f est une formule valide, on note $\models f$

- $\{f_1, .., f_n\} \models f$ équivaut à $\models (f_1 \wedge .. \wedge f_n) \Rightarrow f$

  ou $\models \neg (f_1 \wedge .. \wedge f_n) \vee f$

  ou, si les formules sont **closes**, $f_1 \wedge .. \wedge f_n \wedge \neg f$ est inconsistante (preuve par réfutation ou par l'absurde)

Système formel et preuve

- La notion de *conséquence logique* oblige, pour vérifier qu'une formule est satisfaite par des hypothèses, à utiliser un domaine et à assigner des valeurs de vérités : il s'agit d'une méthode sémantique

- La notion de *démonstration* (ou de preuve) est purement syntaxique : on applique formellement des règles pour passer mécaniquement des hypothèses à la formule
  - on introduit un cadre formel pour les démonstrations

- Un système formel S est constitué de :
  - F un ensemble de formules
  - A un ensemble d'axiomes $A \subset F$
  - un ensemble fini de règles de déduction valides

Déduction

- Une preuve dans un système formel S est une suite finie d'énoncés A1, .. An telle que pour tout i, A_i est un axiome de S ou une conséquence des A_j (j<i) par application d'une règle de déduction.

- Un théorème de S est le dernier énoncé d'une preuve. Si A est un théorème, on note $\vdash A$

- Une formule A est déductible d'un ensemble de formules $\{f_1, .., f_n\}$ ssi il existe une suite finie $A_1, .., A_n$ d'énoncés telle que $A_n =$ A et pour tout i<n, A_i est un axiome ou $A_i \in \{f_1, .., f_n\}$ ou A_i découle des A_j (j<i) par application d'une règle de déduction.
  On note $\{f_1, .., f_n\} \vdash A$

Règles de déduction

- Modus ponens : $\{(f \Rightarrow g), f\} \vdash g$

- Modus tollens : $\{(f \Rightarrow g), \neg g\} \vdash \neg f$

- Syllogisme : $\{(f \Rightarrow g), (g \Rightarrow h)\} \vdash (f \Rightarrow h)$

- Généralisation : $f \vdash \forall x\, f$

- ...

- *Propriété* : F et G étant deux formules, $\{F\} \vdash G$ si et seulement si $F \Rightarrow G$ est un théorème ($\vdash (F \Rightarrow G)$)

Complétude et correction (1/2)

Théorie des modèles	Théorie de la démonstration
Interprétation sémantique sur un domaine	*Interprétation syntaxique*
Tables de vérité des connecteurs et prédicats	*Axiomes, règles d'inférence*
Tautologie	*Théorème*
Conséquence $\models$	*Déduction* $\vdash$

- Un système est complet ssi $\models$ g implique $\vdash$ g (on peut démontrer toutes les tautologies)
- un système est correct ssi $\vdash$ g implique $\models$ g (tous les théorèmes sont des tautologies)

Complétude et correction (2/2)

- *Théorème* : le calcul des prédicats est correct et complet (Gödel, 1929)
- En particulier pour le système suivant (dit système minimal)
- *axiomes du calcul propositionnel* (a,b et c étant des formules)
  - $a \Rightarrow (b \Rightarrow a)$
  - $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$
  - $(\neg b \Rightarrow \neg a) \Rightarrow (a \Rightarrow b)$
- *axiomes du calcul des prédicats* (a,b étant des formules et x une variable)
  - $\forall x\ a(x) \Rightarrow a(t)$
  - $(a \Rightarrow b) \Rightarrow (a \Rightarrow \forall x\ b)$
- *régles* : *modus ponens* et *généralisation*

Décidabilité

- Un système est décidable s'il existe un algorithme permettant de décider à coup sur si une formule est vraie ou fausse

- *Théorème* : le calcul des propositions est décidable (méthode des tables de vérité)

- *Théorème* : le calcul des prédicats est indécidable (Church 1936)
  - en fait il est semi-décidable : on peut toujours prouver en un temps fini qu'une formule est vraie (complétude) mais pas qu'une formule est fausse

Principe de résolution

- Le principe de résolution (Robinson 1965) est une règle de déduction en *logique propositionnelle* :

$$A \vee B, \neg A \vee C \vdash B \vee C$$

- Le principe de résolution est *valide*
  - il faut montrer $(A \vee B) \wedge (\neg A \vee C) \Rightarrow B \vee C$
  - Si A est vrai, alors C est vrai donc $B \vee C$ aussi
  - Si A est faux, alors B est vrai et donc $B \vee C$ aussi

- Pour utiliser le principe de résolution, il faut identifier un littéral et sa négation dans une formule (A et ¬A) et à fabriquer la formule résultante ne dépendant plus du littéral

- Le principe de résolution permet, par réfutation, de démontrer une formule à partir d'un ensemble de formules, si toutes ces formules sont sous forme clausales

Résolution et réfutation

- Procédure de résolution par réfutation pour prouver une formule F sous forme clausale à partir d'un ensemble de formules clausales $\{F_1,..,F_n\}$
  - on prend la négation de F
  - on prouve par résolution que $\{F_1,..,F_n, \neg F\}$ est inconsistant en calculant les résultantes jusqu'à obtenir la clause vide

$$\mathbf{A} \vdash_{reso} \mathbf{B} \text{ si et seulement si } \mathbf{A} \wedge \{\neg \mathbf{B}\} \vdash_{reso} \square$$

- *Théorème* :
  - si un ensemble de clauses est insatisfiable, alors il admet une réfutation par résolution (complétude)
  - si un ensemble de clauses admet une réfutation par résolution, il est insatisfiable (correction)

Exemple de résolution

- Formule propositionnelle à démontrer :
  $\{p \Rightarrow r, q \Rightarrow r\} \models (p \vee q) \Rightarrow r$
- La négation de $(p \vee q) \Rightarrow r$ est $\neg (\neg (p \vee q) \vee r) \equiv (p \vee q) \wedge \neg r$
- $\{\neg p \vee r, \neg q \vee r , p \vee q, \neg r\}$ est l'ensemble de clauses de départ
- $\neg p \vee r$ et $\neg r$ sont résolues en $\neg p$
- $\neg q \vee r$ et $\neg r$ sont résolues en $\neg q$
- $\neg p$ et $p \vee q$ sont résolues en q
- q et $\neg q$ sont résolues en $\square$

Résolution et clauses de Horn

- La résolution par réfutation n'est pas toujours efficace si on ne choisit pas les bonnes clauses
  - en particulier, dans le cas général, il n'est pas plus efficace que les méthodes sémantiques consistant à construire des interprétations (algorithme de Quine, de Davis & Putnam)
- Dans un système de démonstration automatique, il faut pouvoir choisir les bonnes clauses pour avoir un calcul efficace
- Solution => utiliser des clauses de Horn

Résolution en calcul des prédicats

- On veut résoudre des formules du calcul des prédicats à l'aide du principe de résolution en calcul propositionnel
- Pour utiliser la réfutation par résolution en calcul des prédicats, il faut transformer les formules en clauses
  - **Première étape** : on transforme la formule en formule normale conjonctive (conjonction de clauses)
  - **Deuxième étape** : on tranforme les formules normales conjonctives en formules normales prénexes (avec les quantificateurs en tête)
  - **Troisième étape** : on *skolémise* la formule obtenue pour éliminer les quantificateurs

Forme normale conjonctive

- Forme normale conjonctive : conjonction de disjonctions

- Règles de transformation en forme normale conjonctive :
  - on transforme les $\Rightarrow$ par équivalence $p \Rightarrow q \equiv \neg p \vee q$
  - on accole les négations aux atomes en utilisant $\neg \neg F \equiv F$ et les lois de Morgan
  - on utilise la distributivité de $\wedge$ et $\vee$ pour obtenir une conjonction de clauses
  - on renomme les variables si nécessaire

- *Théorème* : toute formule admet une forme normale conjonctive équivalente

Forme prénexe

- Forme prénexe : les quantificateurs sont en tête de formule

- Règles pour transporter les quantificateurs en tête de formule :

  $\neg \forall x\, F \equiv \exists x \neg F$ $\qquad$ $\neg \exists x\, F \equiv \forall x \neg F$

  $\forall x \forall y\, F \equiv \forall y \forall x\, F$ $\qquad$ $\exists x \exists y\, F \equiv \exists y \exists x\, F$

  $\forall x\, F \wedge \forall x\, H \equiv \forall x\, (F \wedge H)$ $\qquad$ $\exists x\, F \vee \exists x\, H \equiv \exists x\, (F \vee H)$

  - Si H ne contient aucune occurence de x :

  $(\forall x\, F) \vee H \equiv \forall x\, (F \vee H)$ $\qquad$ $(\exists x\, F) \wedge H \equiv \exists x\, (F \wedge H)$

  $\forall x\, F \equiv F$ $\qquad$ $\exists x\, F \equiv F$

  - Renommer les variables si besoin est

- *Théorème* : toute formule admet une forme prénexe équivalente

Exemple

- Mise sous forme normale prénexe de la formule
  $\forall x\ p(x) \wedge \exists y\ q(y) \Rightarrow \exists y\ (p(y) \wedge q(y))$

- Suppression de $\Rightarrow$:
  - $\neg (\forall x\ p(x) \wedge \exists y\ q(y)) \vee \exists y\ (p(y) \wedge q(y))$

- Renommage des variables :
  - $\neg (\forall x\ p(x) \wedge \exists y\ q(y)) \vee \exists z\ (p(z) \wedge q(z))$

- Transfert de la négation :
  - $(\exists x \neg p(x) \vee \forall y \neg q(y)) \vee \exists z\ (p(z) \wedge q(z))$

- Déplacement des quantificateurs :
  - $\exists x\ \forall y\ \exists z\ (\neg p(x) \vee \neg q(y) \vee (p(z) \wedge q(z)))$

- Forme normale :
  - $\exists x\ \forall y\ \exists z\ ((\neg p(x) \vee \neg q(y) \vee p(z)) \wedge (\neg p(x) \vee \neg q(y) \vee q(z))$

Forme de Skolem

- On élimine les quantificateurs existentiels :
  - remplacer toute variable quantifiée existentiellement par une fonction ayant pour arguments les variables quantifiées universellement précédant la première variable
  - cette fonction est celle qui prend sur le domaine d'interprétation la valeur qui rend la formule vraie

- *Théorème* : si F est une formule, il existe F' forme de Skolem de F et $\models$ F ssi $\models$ F' (ce n'est pas une équivalence logique!)

- Une fois les quantificateurs existentiels supprimés, toutes les variables restantes sont quantifiées universellement (dans une formule close), on peut donc supprimer les quantificateurs universels

Exemple

- Skolémisons la formule normale prénexe
  $\exists x \, \forall y \, \exists z \, ((\neg p(x) \lor \neg q(y) \lor p(z)) \land (\neg p(x) \lor \neg q(y) \lor q(z))$
- La variable z est tranformée en f(y)
  - $\exists x \, \forall y \, ((\neg p(x) \lor \neg q(y) \lor p(f(y))) \land (\neg p(x) \lor \neg q(y) \lor q(f(y)))$
- La variable x est transformée en g (fonction d'arité nulle ou constante)
  - $\forall y \, ((\neg p(g) \lor \neg q(y) \lor p(f(y))) \land (\neg p(g) \lor \neg q(y) \lor q(f(y)))$
- Démontrer une formule c'est prouver que sa négation est inconsistante ou que la forme de Skolem de sa négation est inconsistante
- On peut donc se limiter à travailler sur des formes de Skolem

Théorème de Herbrand (1/4)

- Pour appliquer le principe de résolution à des formes de Skolem, il faut donner des valeurs aux variables universelles
- Impossible en pratique de résoudre une formule pour toutes les valeurs possibles des variables sur un domaine
- *Intérêt du théorème de Herbrand* : Quand on a une formule sous forme de Skolem, on peut se limiter pour étudier sa satisfiabilité à son univers de Herbrand
- Termes de base et atomes de base d'un ensemble de clauses E
  - un terme de base est un terme qui ne contient pas de variable
  - un atome de base est un atome qui ne contient pas de variable

Théorème de Herbrand (2/4)

- Univers de Herbrand d'un ensemble de clauses E : l'univers de Herbrand de E est l'ensemble des termes de base que l'on peut construire à partir des symboles de fonctions et des constantes qui apparaissent dans E

- *Exemple : l'univers de Herbrand de l'ensemble* $\{p(f(x)) \Rightarrow q(a), r(g(x))\}$ *est* $\{a, f(a), g(a), f(f(a)), f(g(a)), g(f(a)), \ldots\}$

- Base de Herbrand d'un ensemble de clauses E : la base de Herbrand de E est l'ensemble des atomes de base qui peuvent être construits à partir des symboles de prédicats de E appliqués aux termes de l'univers de Herbrand de E

- *Exemple : la base de Herbrand de l'ensemble* $\{p(f(x)) \Rightarrow q(a), r(g(x))\}$ *est* $\{p(a), q(a), r(a), p(f(a)), q(f(a)), r(f(a)), p(g(a)), q(g(a)), r(g(a)), \ldots\}$

Théorème de Herbrand (3/4)

- Interprétation de Herbrand : l'ensemble de définition est l'univers de Herbrand. Une interprétation de Herbrand d'un ensemble E de clauses est obtenue en remplaçant les variables de E par des éléments de l'univers de Herbrand de E

- Une interprétation de Herbrand est une interprétation mais pas le contraire.

- *Exemple : une interprétation de Herbrand de l'ensemble* $\{p(f(x)) \Rightarrow q(a), r(g(x))\}$ *est* $\{p(f(a)) \Rightarrow q(a), r(g(f(a)))\}$

- Modèle de Herbrand d'un ensemble de clauses E : c'est une interprétation de Herbrand de E qui est un modèle de E

Théorème de Herbrand (4/4)

- Théorème de Herbrand (Herbrand 1929) : un ensemble de clauses E est insatisfiable si et seulement si il existe un ensemble fini d'interprétations de Herbrand de E qui soit insatisfiable

- *Conséquence* : montrer une formule sous forme clausale revient à trouver une interprétation de Herbrand qui soit insatisfiable

- Pour montrer qu'une formule F est valide :
  - on construit F', la forme normale de Skolem de sa négation
  - on trouve une interprétation de Herbrand
  - on montre par résolution que cette interprétation est insatisfiable
  - => F' est donc insatisfiable et donc F est valide

- Le principe de résolution doit être étendu au calcul des prédicats à travers le mécanisme d'unification

Substitution

- *Exemple : soient les clauses $C_1 = p(x) \vee q(x)$ et $C_2 = \neg p(f(y)) \vee r(y)$*
  - *on ne peut appliquer la résolution, car aucun littéral de C_1 n'est la négation d'un littéral de C_2 ou l'inverse*
  - *on voudrait pouvoir substituer f(y) à x dans C1, ce qui donnerait par résolution $q(x) \vee r(y)$*

- Une substitution consiste à remplacer un nombre fini de variables par des termes. On note $\{t_1/v_1, .. , t_n/v_n\}$ la substitution qui remplace toute variable v_i par le terme t_i.

- L'application d'une substitution S à un ensemble de clauses E est appelé instance de E selon S

Unificateur

- Composition de substitution : la composition de deux substitutions s_1 et s_2, notée $s_1 o\ s_2$, est obtenue en 3 étapes
  - appliquer s_2 aux termes de s_1
  - retirer de s_2 les couples t_i/v_i tels que v_i est une variable de s_1
  - rassembler les couples obtenues en 1 et 2

- *<u>Exemple</u> : s_1 = {f(y)/x , z/y} et s_2 = {a/x , b/y , y/z}*
  - *la première étape donne {f(b)/x , y/y}*
  - *la deuxième donne {y/z}*
  - *la troisième étape donne {f(b)/x , y/z} (on supprime y/y qui ne change rien)*

- Unificateur : une substitution S unifie un ensemble de clauses E = {c_1, .. , c_n} si S(c_1) = .. = S(c_2)

- *<u>Exemple</u> : {f(a)/x , a/y} unifie {p(a,x), p(a,f(y))}*

Unificateur le plus général

- Unificateur le plus général : l'unificateur le plus général d'un ensemble de clauses E est un unificateur U de E tel que pour tout autre unificateur V de E, il existe une substitution S telle que V = S o U

- Cet unificateur le plus général n'existe pas forcément (pas plus qu'un unificateur) et s'il existe, il n'est pas forcément unique

- Trouver un unificateur le plus général permet d'appliquer le principe de résolution à des clauses issues de formules du premier ordre

Algorithme d'unification

- Algorithme d'unification
  - Données : deux expressions E_1 et E_2
  - **si** E_1 ou E_2 est un atome **alors** échanger les données de façon à ce que E_1 soit un atome et passer à 2
    sinon passer à 3
  - **si** E_1 et E_2 sont identiques **alors** retourner Ø
    sinon si E_1 est une variable
    si E_1 a une occurence dans E_2 **alors** retourner échec
    sinon retourner $\{E_2/E_1\}$
    sinon si E_2 est une variable **alors** retourner $\{E_1/E_2\}$
    sinon retourner échec
  - F_1 := le premier élément de E_1, T_1 := le reste de E_1 F_2 := le premier élément de E_2, T_2 := le reste de E_2
  - U_1 := unification(F_1,F_2)
  - **si** U_1 = échec **alors** retourner échec
    sinon $G_1 := U_1(T_1)$, $G_2 := U_1(T_2)$
  - U_2 := unification(G_1,G_2)
  - **si** U_2 = échec **alors** retourner échec **sinon** retourner $U_1 o U_2$

Récapitulatif

- Pour démontrer qu'une formule F du calcul des prédicats peut être déduite d'un ensemble E de clauses
  - on procède par réfutation en créant G = ¬ F
  - la skolémisation de G permet de ramener le problème de l'inconsistance de G U E à celui de l'inconsistance d'un ensemble de clauses $\{C_1,..,C_n\}$ U E
  - le théorème de Herbrand permet de ramener la démonstration de l'inconsistance de $\{C_1,..,C_n\}$ U E à la découverte d'une instanciation des variables dans l'univers de Herbrand qui rende l'ensemble de clauses insatisfiable
  - le principe de résolution permet de montrer l'insatisfiabilité de l'instance construite
- En Prolog, on n'a que des clauses de Horn
  - La skolémisation est inutile mais elle garantit que ce qu'on écrit en Prolog est quasiment aussi expressif que la logique des prédicats
  - Le mécanisme de Prolog consiste à construire une instanciation de la clause but et de clauses de la base de connaissance tout au long d'une procédure de résolution par réfutation