L1 - UE Logique 1

Université de Toulouse-Paul Sabatier/ IRIT

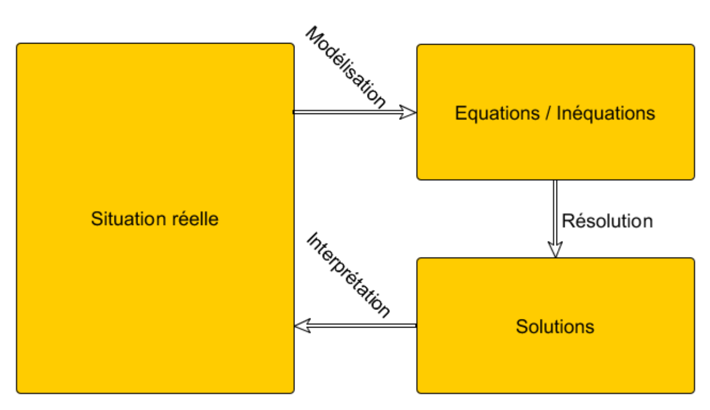
Année universitaire 2018-19

Plan

1. Motivation
   - Problématique et histoire
   - Applications en informatique
2. Logique des propositions
3. Logique des prédicats

Problématique

Modélisation mathématique



Problématique

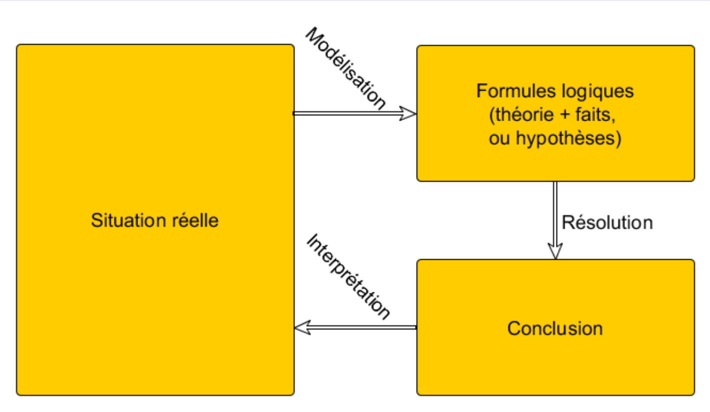
Modélisation mathématique

Comment gérer :

- Une prise en charge médicale (PCM) est invasive si et seulement si elle nécessite une injection ou une sonde respiratoire
- Si une PCM ne nécessite pas d'injection alors si l'électrocardiogramme (ECG) n'est pas normal alors elle nécessite une sonde respiratoire
- S'il y a présence d'ondes P alors l'ECG n'est pas normal
- Conclusion ?

Problématique

Modélisation logique



Problématique

Modélisation logique

Modélisation logique :

- L'ensemble d'hypothèses $\{i \leftrightarrow q \vee s, \neg q \rightarrow \neg n \rightarrow s, p \rightarrow \neg n\}$
- entraîne-t-il la conclusion : $\neg i \rightarrow \neg p$

Vous apprendrez à :

- Modéliser (formaliser)
- Résoudre
- Interpréter

Préambule : un peu de vocabulaire

- Un **énoncé** est une phrase pouvant être vraie ou fausse :
  - "Vous aurez Logique lundi 30/01 ou mardi 31/01"
  - "La vitesse de la lumière est constante"
  - "La fonction $f(x) = 2.\cos(x)$" est continue et dérivable sur $\mathbb{R}$
- Une **proposition** est un énoncé atomique/élémentaire (ne contenant pas de sous-proposition) :
  - "Vous aurez Logique lundi 30/01"
  - "La vitesse de la lumière est constante"
  - $2^{137107291} - 1$ est un nombre premier.

Préambule : un peu de vocabulaire

- Une théorie est un ensemble d'énoncés
- Un jugement est l'affirmation d'un lien de conséquence entre une théorie (les hypothèses) et un énoncé (la conclusion) :

Préambule : un peu de vocabulaire

Exercice 1 :

- Ce qui est rare est cher
- Les lingots d'or bon marché sont rares

Donc :

- Les lingots d'or bon marché sont chers

Quel est le problème ? Un jugement erroné ? Absence de lien de conséquence ? Erreur de modélisation ? Autre ?

Histoire de la logique des prédicats

Le syllogisme d'Aristote

Instance :

- Prémisse majeure :
  Tous les hommes sont mortels
- Prémisse mineure :
  Tous les Grecs sont des hommes
- Conclusion :
  Tous les Grecs sont mortels

En général :

- Tous les B sont des C
- Tous les A sont des B
- Tous les A sont des C

Histoire de la logique des prédicats

La syllogistique

- Aucun M n'est P, or tout S est M
  donc aucun S n'est P ;
- Tout P est M, or quelque [1] S n'est pas M
  donc quelque S n'est pas P ;
- Tout P est M, or tout M est S
  donc quelque S est P ;
- ...

(on apprenait tout cela par cœur au Moyen-Âge)

1. Quelque, au singulier : au moins un

Histoire de la logique des prédicats

La syllogistique

- Chez Aristote, les propositions sont analysées en quantificateur (tous, aucun, certains, quelque(s),...) + sujet + verbe + complément.
- Ces propositions analysées ont une forme simple.
- On parlera de logique des prédicats pour des formes plus complexes.

Observation essentielle :

- On peut reconnaître un argument valide par sa forme (sa syntaxe)
- ...sans regarder la signification des mots (leur sémantique)

Histoire de la logique des prédicats

Les syllogismes d'Aristote

Structure générale : deux prémisses, une conclusion

Traduction moderne des quatre formes d'énoncé :

- Tout A est B : $A \subseteq B$
- Aucun A n'est B : $A \cap B = \{\}$
- Quelque A est B : $A \cap B \neq \{\}$
- Quelque A n'est pas B : $A \not\subseteq B$

Comment en vérifier la validité ?

- P_1 : Quelque B est C
- P_2 : Tout A est B
- C : Quelque A est C

(Explication intuitive type diagramme de Venn)

Syllogismes

Exercice 2 : Lesquels de ces syllogismes sont valides ?

1. Aucun chat n'est humain. Tout humain est un bipède. Donc aucun chat n'est bipède.
2. Les pies ont des plumes. Or les chats n'ont pas de plumes. Donc les chats ne sont pas des pies.
3. Les pies sont des oiseaux. Les pies sont bicolores. Donc quelque oiseau est bicolore.
4. Tout diptère est ailé. Or quelque insecte n'est pas ailé. Donc quelque insecte n'est pas un diptère.

Histoire de la logique des propositions

- Chez les stoïciens, les propositions ne sont pas analysées, elles forment un bloc.
- On parlera de logique des propositions.
- Née à cause de paradoxes logiques : “Cette phrase est fausse” $\Rightarrow$ vraie ou fausse ?
- Les connecteurs : disjonction (ou), conjonction (et), négation (non), implication (si. . . alors) relient des propositions (non-analysées), pour former des énoncés.
- Quelques uns des "indémontrables"[2] :
  - Si A alors B. Or A. Donc B. (Modus Ponens)
  - Si A alors B. Or non B. Donc non A. (Modus Tollens)
  - A ou B. Or non A. Donc B.
  - A et B. Donc A.

2. Ce sont des jugements qui sont admis comme étant valides.

Histoire de la logique des propositions

Exemples :

- Si j'ai de la fièvre alors je suis malade. Or j'ai de la fièvre. Donc je suis malade.
- S'il pleut alors le sol est mouillé. Or le sol n'est pas mouillé. Donc il ne pleut pas.
- Tu as l'as de pique ou l'as de carreau. Or tu n'as pas l'as de carreau. Donc tu as l'as de pique.

Histoire de la logique des propositions

Important : “Si . . .alors . . .” ne marque pas nécessairement une relation de cause à effet mais une simple simultanéité.

- La fièvre ne cause pas la maladie (c’est même l’inverse)
- mais la fièvre évoque toujours la maladie, la fièvre est une condition suffisante : il suffit d’avoir la fièvre pour être déclaré malade
- alors qu’on peut être malade sans fièvre, ça n’est pas une condition nécessaire : il n’est pas nécessaire d’avoir de la fièvre pour être déclaré malade.

Histoire de la logique des propositions

1. Réciproquement : être malade, est-ce une condition nécessaire pour avoir de la fièvre ? Une condition suffisante ?
2. De manière générale, si A est CN de B, qu'est B par rapport à A ?
3. Et si A est CS de B, qu'est B par rapport à A ?
4. Pas de fumée sans feu et pas de feu sans fumée : statut de "fumée" vs. "feu" ?

Histoire de la logique des propositions

Condition nécessaire vs. suffisante

Quelle est le rôle de "j'ai un bon tarif" dans ces énoncés :

1. Je prends l'avion si j'ai un bon tarif ;
2. Je prends l'avion seulement si j'ai un bon tarif ;
3. Je prends l'avion si et seulement si j'ai un bon tarif.

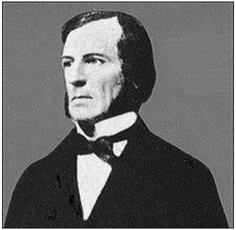
Attention : le langage courant ne fait pas toujours ces distinctions, cela est source de sous-entendus et donc d'ambigüités.

Histoire de la logique : Leibniz

Leibniz (1646 - 1716)

- Codage binaire des nombres
- Construction de l'un des premiers calculateurs (mécanique, décimal)
- Développement du calcul infinitésimal (en concurrence avec Newton)
- Ars combinatoria : l'art de dériver des vérités de manière calculatoire, basé sur
  - une characteristica universalis, un langage mathématique non-ambigu
  - un calculus ratiocinator, un calcul / une machine manipulant la characteristica

Histoire de la logique : Boole

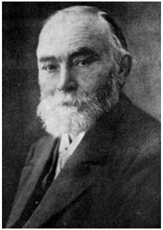


Boole (1815 - 1864)

Livre : An Investigation of the Laws of Thought

- Traitement algébrique de la logique propositionnelle
- Procédure de décision pour la logique propositionnelle

Histoire de la logique : Frege



Frege (1848 - 1925)

- Fondateur de la logique “moderne” :
  - les connecteurs “essentiels” de la logique des propositions : $\longrightarrow$ et $\neg$
  - les quantificateurs de la logique des prédicats
  - un calcul formel

Histoire de la logique : Frege

"Begriffsschrift" de Frege (1879)

Les jugements

$\vdash a \longrightarrow b \longrightarrow a$

et

$\vdash (c \longrightarrow b \longrightarrow a) \longrightarrow ((c \longrightarrow b) \longrightarrow (c \longrightarrow a))$

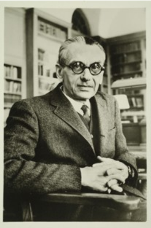
- Distinction entre
  - formule : représente une proposition (qui peut être vraie ou fausse)
  - jugement : formule dont on constate la vérité (dans un calcul donné)
- Notation deux-dimensionnelle qui a laissé des traces :
  - négation ($\neg$)
  - jugement ($\vdash$)

Histoire de la logique : Russell

Russell (1872 - 1970)

- Livre : Principia Mathematica (1910 - 1913, avec A. N. Whitehead)
  But : formalisation des mathématiques à partir de quelques notions élémentaires
- Découverte d'un paradoxe (1903) dans la théorie des ensembles de Cantor
  $\rightsquigarrow$ crise des fondements de la mathématique :
  Consistance des axiomes ?
- (En plus : prix Nobel de littérature, emprisonné suite à des campagnes contre l'armement nucléaire, ...)

Histoire de la logique : Gödel



Gödel (1906 - 1978)

Le cataclysme : "Sur des propositions indécidables de Principia Mathematica" (1931)

- Toute théorie mathématique "suffisamment expressive" est
  - incomplète : il existe des énoncés vrais non-démontrables
  - ou contradictoire
- Surtout, elle ne permet pas de démontrer sa propre cohérence
- $\rightsquigarrow$ échec du programme rationaliste dans la tradition Leibniz - Hilbert

Histoire de la logique - Résumé

Quelques repères dans un paysage vaste :

- Syllogisme d'Aristote : Possibilité de raisonner en s'appuyant sur la syntaxe, sans connaissance de la sémantique
- Leibniz : Raisonnement exécutable par une machine (calcul)
- Forme moderne de la logique (Boole, Frege)
- Le programme rationaliste mis à mal : Russell, Gödel

Pour plus d'information sur le développement de la logique au tournant du 20ème siècle : Jean Van Heijenoort : From Frege to Gödel

Logique des propositions

Exercice 3 :

Lesquels de ces jugements sont valides ?

1. S'il y a du verglas, la route est glissante.
   Si la route est glissante, les camions ne roulent pas.
   Or les camions roulent.
   Donc il n'y a pas de verglas.
2. Quand il ne pleut pas, le sol est sec.
   Quand il n'y a pas d'humidité dans l'air, il ne pleut pas.
   Or l'air est humide.
   Donc le sol n'est pas sec.

Logique des propositions

Exercice 4 :
Quelles propositions utiliserez-vous pour formaliser :

1. Une prise en charge médicale (PCM) est invasive si et seulement si elle nécessite une injection ou une sonde respiratoire
2. Si une PCM ne nécessite pas d'injection alors si l'électrocardiogramme (ECG) n'est pas normal alors elle nécessite une sonde respiratoire
3. S'il y a présence d'ondes P alors l'ECG n'est pas normal
4. Donc si la prise en charge médicale (PCM) n'est pas invasive il n'y a pas présence d'ondes P

Comment le résoudre ? (on y reviendra)

Plan

1. Motivation
   - Problématique et histoire
   - Applications en informatique
2. Logique des propositions
3. Logique des prédicats

Applications de la logique

La logique a d'importantes applications en informatique :

- Intelligence artificielle (cf. conférence S1 : simuler la pensée)
- Sûreté de fonctionnement (cf. conférence S1 : The big bug theory)
- Sécurité

Objectifs de cette section :

- Comprendre le rôle de la logique dans l'informatique
- Gagner une intuition des méthodes mises en œuvre

Application : Intelligence artificielle (1)

But :

- Comprendre mieux le fonctionnement de l'intelligence humaine
- Découvrir les facettes d'un comportement rationnel

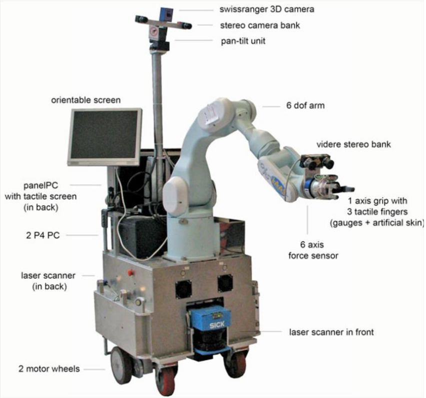
Applications :

- Simuler le comportement humain :
  économie
- Guider l'action humaine :
  aide au diagnostic (médical, informatique,. . .)
- Imiter le comportement humain :
  robotique, jeux classiques, jeux vidéos (personnages virtuels)

Application : Intelligence artificielle (2)

Robotique :

- Contrôle des actions : Comment déplacer un livre d'une chaise sur une table ?
- Contrôle des mouvements : Comment se déplacer d'une salle à une autre ?
- Interaction homme-machine : Comment interpréter une commande en langage naturel ?



Robot Jido (LAAS)

Application : Sûreté de fonctionnement (1)

But : S'assurer du bon fonctionnement du matériel et logiciel de systèmes critiques :

- Transport (avions, trains)
- Nucléaire
- Systèmes médicaux

Contexte :

- Systèmes embarqués de plus en plus sophistiqués
- Exemple : Système fly by wire de l'A330

à lire : Wired: History's worst software bugs

Application : Sûreté de fonctionnement (2)

Explosion de la fusée Ariane 5 (juin 1996) :

- 40 sec. après son lancement
- À bord : 4 satellites de recherche
- Dispersion de 745 tonnes de débris, y compris substances toxiques
- Coût des dégâts : 290 millions d'Euros

Contexte :

- Premier lancement de la fusée
- Lors du développement : Réutilisation de logiciel de l'Ariane 4 (autres caractéristiques de vol, accélération moins forte)
- Cause : dépassement arithmétique

Application : Sûreté de fonctionnement (3)

- Exemple du domaine médical : Therac-25 (entre 1985 et 1987)
  - Lors d'une radiothérapie, 6 personnes reçoivent une surdose massive (facteur 100)
  - Conséquence : 3 décès
  - Causes : Multiples, surtout :
    problème de synchronisation de deux tâches
- Mai 2007, à Toulouse :
  - 145 personnes irradiées au CHU de Rangueil
    "c'est encore une fois une déficience de l'informatique qui serait en cause [...] L'étalonnage était mal fait [...]"
  - lire l'article dans le Parisien

Application : Sécurité

Problème :

- Des virus et vers informatiques infectent des systèmes connectés par Internet
- Exemple : Ver “Slammer” (janvier 2003)
  - infecte 75.000 machines en 30 min
  - entre autres : blocage du système de surveillance de la centrale nucléaire Davis-Besse (Ohio) pendant 5h

Causes de la vulnérabilité :

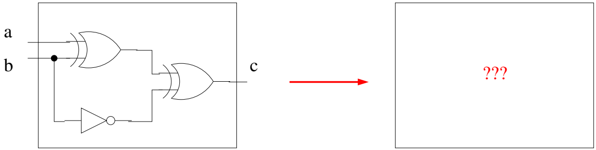
- Langages de programmation de bas niveau (assembleur, C)
- Incompréhension du fonctionnement des protocoles de communication
- Cryptage insuffisant

à lire : Wired: Slammer

Applications : architecture

But : simplification de circuits électroniques
Problème :

- Utiliser moins de composants

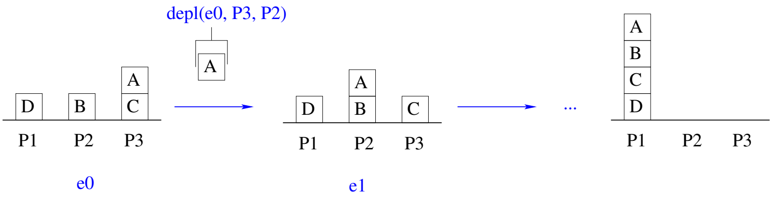


Méthodes : Systèmes à état

Robotique / Planification :

Exemple :

- Un état est caractérisé par des objets A, B, C, D empilés arbitrairement sur trois positions P_1, P_2, P_3
- Opérations : Le robot peut déplacer un seul objet au sommet d'une pile vers le sommet d'une autre pile
- But : empiler A, B, C, D (dans l'ordre) sur P_1
- Solution : séquence d'opérations qui atteignent le but



Méthodes : Systèmes à état

Modélisation :

- Prédicat ternaire `sur` : un bloc est sur un autre bloc / sur une position dans un état
  Exemples : `sur(D, P1, e0)` et `sur(A, B, e1)`
- Fonction ternaire `depl` : convertir un état en déplaçant le sommet d'une position vers une autre
  Exemple : L'état `e1` est : `depl(e0, P3, P2)`

à faire :

- Décrivez entièrement l'état `e1`
- Trouvez la séquence d'opérations menant au but

Méthodes : Systèmes à état

Recherche d'une solution : On peut

- décrire le but par une formule logique :
  "Pour tout état initial, il existe un état final ayant la propriété ..."
- faire une preuve constructive $\rightsquigarrow$ séquence d'opérations

Remarque : Formule et preuve en logique des prédicats

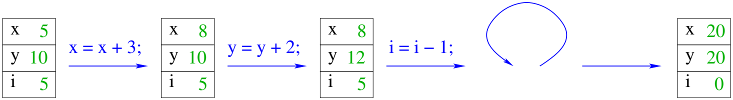
Questions plus générales :

- Comment trouver une solution pour n'importe quel état initial ?
  $\rightsquigarrow$ stratégie de preuve
- Est-ce qu'on peut assurer qu'une solution existe toujours ?
  $\rightsquigarrow$ complétude de la stratégie
- ...aussi si on a seulement deux positions ?

Méthodes : Systèmes à état

Programmes impératifs :

- Un état est caractérisé par un ensemble de variables et de valeurs associées.
- Opérations :
  - Affectations
  - Séquences, structures de contrôle, boucles



Méthodes : Systèmes à état

Pré- /Postconditions :

```
{ y = 2 * x ∧ x ≥ 0 }
i = x;
while (i > 0) {
  x = x + 3;
  y = y + 2;
  i = i - 1;
  }
{ y = x }
```

Correction du programme :

- Est-ce que le programme transforme tout état qui satisfait la précondition en un état qui satisfait la postcondition ?
- Vérification à l'aide d'un invariant : $i \geq 0 \wedge \; y - x = i$

$\rightsquigarrow$ plus de détails dans le cours “Algo. Prog.”

Résumé

La logique est essentielle dans des domaines tels que

- l'intelligence artificielle
- la sûreté de fonctionnement et la sécurité

La logique permet de décrire

- l'équivalence fonctionnelle et comportementale (exemple : circuit)
- les propriétés des systèmes de transition :
  - exemple planification :
    Donné : État initial, état final.
    But : Recherche de séquence d'actions
  - exemple programme impératif :
    Donné : Séquence d'actions (le programme)
    But : Vérification de rapport entre un état initial et un état final

Plan

1. Motivation
2. Logique des propositions
   - Syntaxe (langage)
   - Sémantique (théorie des modèles)
   - Résolution
3. Logique des prédicats

Langages logiques (1)

Les langages logiques sont :

- des abstractions du langage naturel ou mathématique :
  - Lang. nat. : “Il fait nuit, et la lumière est allumée”
  - Logique : $n \wedge l$
  - avec : n associé à "il fait nuit" et l à "la lumière est allumée"
- ...qui ne permettent pas d’exprimer certaines nuances :
  - Lang. nat. : “Il fait nuit mais la lumière est allumée”
  - Logique : $n \wedge l$
- ...mais plus précis :
  - Lang. nat. : “Je suis au bureau ou je suis dans le jardin et je lis un livre”
  - Logique : $(b \vee j) \wedge l$ ou $b \vee (j \wedge l)$
  - avec : b associé à "je suis au bureau", j associé à "je suis dans le jardin" et
    l associé à "je lis un livre"

Langages logiques (2)

La logique des propositions LProp

- est une logique très simple
- fait une abstraction très grossière

Elle permet difficilement d'exprimer

- des rapports entre des individus en nombre infini ou non borné :
  "entre deux nombres, il y a un troisième nombre"
  $\rightsquigarrow$ logique des prédicats
- des rapports temporels :
  "S'il fait nuit, la lumière va s'allumer"
  $\rightsquigarrow$ logiques temporelles
- des souhaits, obligations, possibilités, . . . :
  "Je pense que s'il fait nuit, la lumière devrait être allumée"
  $\rightsquigarrow$ logiques modales

Abstractions de la Lprop (1)

Formaliser en logique des propositions consiste à

- associer des variables propositionnelles à des propositions [3].
  n associé à "il fait nuit", l à "la lumière est allumée"
- les relier par des connecteurs logiques :
  - “Il fait nuit, et la lumière est allumée” : $n \wedge l$
  - “Il fait nuit, ou la lumière est allumée” : $n \vee l$ (sens inclusif)
  - “S’il fait nuit, alors la lumière est allumée” : $n \longrightarrow l$
  - “Il ne fait pas nuit” : $\neg n$
- pour constituer des formules associées à des énoncés [3].

3. v. def. diapo 7

Abstractions de la LProp (2)

Inversion de l'ordre des connecteurs :

- "Je vais à la plage s'il fait beau"
  (a la même signification que "S'il fait beau, je vais à la plage")
  en logique : $b \longrightarrow p$
  avec : b associé à "il fait beau" et p à "je vais à la plage"

Attention à l'ambiguïté du langage naturel :

- "Je vais au cinéma ou à la plage s'il fait beau et pas froid"
  1. $(b \wedge \neg f) \longrightarrow (c \vee p)$
  2. $c \vee ((b \wedge \neg f) \longrightarrow p)$

Formules de la LProp

Ingrédients

- Des variables propositionnelles, des connecteurs, des parenthèses.
  Pour les variables propositionnelles on prend souvent $\{p, q, r \ldots\}$ ou des mnémoniques (f pour fièvre, ...).

Formules de la LProp

Ingrédients

- Des variables propositionnelles, des connecteurs, des parenthèses.
  Pour les variables propositionnelles on prend souvent $\{p, q, r \ldots\}$ ou des mnémoniques (f pour fièvre, ...).

Le "bottom" et le "top"

On utilisera en plus deux symboles particuliers :

$\perp$ ("bottom") associé à la proposition "toujours fausse",

et $\top$ ("top") à la proposition "toujours vraie".

Formules de la LProp

Recette

- À l'aide de ces ingrédients, en liant les propositions par des connecteurs, on obtient des formules, qui sont associées à des énoncés.
- L'ensemble des propositions est noté *PROP*, et l'ensemble des formules *FORM*.
- NB : les propositions étant elles-mêmes des formules, $PROP \subseteq FORM$.

$\Rightarrow$ parallèle avec constantes (nombres), variables et opérations pour former des expressions arithmétiques.

Formules de la LProp

Liste des connecteurs utilisés

- $\wedge$ (conjonction, se lit "et") ;
- $\vee$ (disjonction inclusive, se lit "ou") ;
- $\oplus$ (disjonction exclusive, se lit "ou-exclusif" ou "xor") ;
- $\longrightarrow$ (implication, se lit "si. . . alors. . . ") ;
- $\leftrightarrow$ (équivalence, se lit "si et seulement si" ou "équivaut") ;
- $\neg$ (négation, se lit "non").

Formules de la LProp

Sous-formules

- Une formule est constituée de formules plus petites (ou de propositions) liées par un connecteur. Ainsi, $c \vee ((b \wedge (\neg f)) \longrightarrow p)$ est constituée de : c et de $((b \wedge (\neg f)) \longrightarrow p)$ reliées par un $\vee$. À son tour, $((b \wedge (\neg f)) \longrightarrow p)$ est constituée de $(b \wedge (\neg f))$ et de p reliées par un $\longrightarrow$.

Formules de la LProp

Parenthèsage

- c et $((b \wedge (\neg f)) \longrightarrow p)$ sont appelées sous-formules immédiates de $c \vee ((b \wedge (\neg f)) \longrightarrow p)$;

Formules de la LProp

Parenthèsage

- c et $((b \wedge (\neg f)) \longrightarrow p)$ sont appelées <u>sous-formules immédiates</u> de $c \vee ((b \wedge (\neg f)) \longrightarrow p)$;
- c et $((b \wedge (\neg f)) \longrightarrow p)$ mais aussi :
  $c \vee ((b \wedge (\neg f)) \longrightarrow p)$, $(b \wedge (\neg f))$, p, $(\neg f)$, b et f
  sont simplement appelées <u>sous-formules</u> de
  $c \vee ((b \wedge (\neg f)) \longrightarrow p)$;
- Attention, $p \vee q$ n'est pas une sous-formule de $p \vee (q \vee r)$

Formules de la LProp

Parenthèsage

- En principe, chaque connecteur introduit une paire de parenthèses permettant d'identifier les sous-formules sur lesquelles il s'applique. Ce qui permet de distinguer entre $(b \longrightarrow (c \vee p))$ et $((b \longrightarrow c) \vee p)$;
- On peut omettre certaines d'entre elles en jouant sur la priorité des connecteurs de façon non-ambigüe : on doit pouvoir les restituer.

Conventions syntaxiques

On peut omettre des parenthèses selon les conventions suivantes :

- Associativité : Les opérateurs binaires associent à droite :
  - $A \wedge B \wedge C$ est égal[4] à $(A \wedge (B \wedge C))$
  - $A \longrightarrow B \longrightarrow C$ est égal à $(A \longrightarrow (B \longrightarrow C))$
- Priorité : La priorité décroissante des opérateurs est $\neg, \wedge, \vee, \longrightarrow$.
  Exemples :
  - $(A \wedge B \vee C) = ((A \wedge B) \vee C)$
  - $(\neg A \vee B) = ((\neg A) \vee B)$
  - $(A \wedge B \longrightarrow A \vee B) = ((A \wedge B) \longrightarrow (A \vee B))$

4. c'est juste qu'on ne voit pas les parenthèses omises, mais elles y sont

Arbres syntaxiques

Plusieurs représentations de la même formule : Arbre syntaxique, représentation textuelle : parenthésage maximal ou minimal.

Langage

Exercice 5 :

- Mettre en parenthésage minimal les formules suivantes :
  - $(((p \vee q) \wedge (s \longrightarrow (\neg r))) \longrightarrow (\neg s))$
  - $((\neg q) \longrightarrow (r \wedge (t \longrightarrow (\neg q))))$
- Mettre en parenthésage maximal les formules suivantes :
  - $\neg(p \longrightarrow \neg p \wedge q) \wedge q \longrightarrow \neg r$
  - $p \longrightarrow q \wedge s \vee p \longrightarrow \neg q \wedge s$
- Tracez les arbres syntaxiques de ces 4 formules
- Listez les sous-formules immédiates des formules 1 et 2.
- Listez les sous-formules des formules 3 et 4.

Modélisation

Exercice 6 :
Dans les jugements suivants, repérez les propositions, identifiez les hypothèses et la conclusion, puis formalisez chacune d'elles (sous forme d'ensemble de formules pour les hypothèses, conclusion à part) :

1. Quand j'ai de la fièvre, je suis malade. Or j'ai de la fièvre. Donc je suis malade.
2. Quand j'ai de la fièvre, je suis malade. Or je ne suis pas malade. Donc je n'ai pas de fièvre.
3. Quand j'ai de la fièvre, je suis malade. Donc quand je ne suis pas malade, je n'ai pas de fièvre.
4. Si les poules ont des mamelles alors ce sont des mammifères, or elles n'ont pas de mamelles, donc ce ne sont pas des mammifères.
5. Richard Nixon était quaker et républicain. Les quakers étaient contre la guerre du Vietnam. Les républicains étaient pour. Contradiction !

Modélisation

Exercice 7 :

6. Si les archéoptérix ont des mamelles ce sont des mammifères et sinon, ce sont des oiseaux ; donc ce sont des mammifères ou des oiseaux.
7. Les lézards sont poïkilothermes ou ce sont des mammifères, or ce ne sont pas des mammifères, donc ils sont poïkilothermes.
8. Quand j'ai de la fièvre je suis malade, quand j'ai des rougeurs je suis malade aussi. Donc si je ne suis pas malade je n'ai ni fièvre, ni rougeurs.
9. Quand j'ai de la fièvre je suis malade, quand j'ai des rougeurs je suis malade aussi. Donc si je suis malade mais que je n'ai pas de fièvre je n'ai pas de rougeurs.
10. Quand il neige il fait froid, quand il y a du verglas il fait froid, or il y a du verglas ou de la neige. Donc on n'est pas en été, car en été il ne fait pas froid.

Modélisation

Exercice 8 :

11. Après une course hippique entre chevaux noir et blanc, le cheval gagnant fut celui qui avait une robe noire et une crinière blanche. Il s'agissait de PetitGris, de Qristal ou de Rikita. Or PetitGris a une crinière noire et une queue blanche, Qristal a une robe noire et une queue blanche, et enfin Rikita a une robe noire. Enfin le gagnant avait sa crinière et sa queue de couleur opposées. C'est donc Rikita le gagnant.

Modélisation

Exercice 9 :

12 Un vol a été commis, on a relevé des traces de chaussures taille 40 et le voleur a été aperçu : il est petit. Parmi les 3 suspects (A, B et C), A est grand et chausse du 40, B est petit et chausse du 45, enfin C est petit et chausse du 40. Le voleur est l'un des trois suspects. Donc C est le voleur.
Question supplémentaire : quelle hypothèse “raisonnable” manque-t-il pour pouvoir conclure à la culpabilité de C ?

Modélisation

Exercice 10 :
Jeu de Tic-Tac-Toe (morpion 3x3) en logique propositionnelle :

- Combien faut-il de variables propositionnelles au minimum (si l'on ne joue pas sur les symétries) pour décrire l'état de la grille à un instant donné ? Lesquelles ?
- Comment peut-on décrire des configurations comme : “le joueur $\circ$ gagne avec la première ligne” ?

Modélisation

Exercice 11 :

Pour intégrer le *No SKRuB* club, il faut (et il suffit de) remplir TOUTES les conditions suivantes :

1. s'il joue de la cornemuse et s'il ne porte pas un kilt alors il n'est pas écossais
2. il joue au rugby ou il porte un kilt
3. s'il joue de la cornemuse et s'il ne joue pas au rugby alors il n'est pas écossais
4. il est écossais ou il joue de la cornemuse
5. s'il est écossais et s'il porte un kilt alors il joue de la cornemuse
6. s'il n'est pas écossais alors il ne joue pas de la cornemuse
7. s'il est écossais et s'il joue au rugby alors il joue de la cornemuse
8. s'il joue au rugby et s'il porte un kilt alors il ne joue pas de la cornemuse

Proposez une formalisation.
Déterminez qui ne peut espérer adhérer à ce club ?

Bonus (Proverbe anglais) : “Un gentleman est un homme qui sait jouer de la cornemuse, mais qui n'en joue pas”

Plan

1. Motivation
2. Logique des propositions
   - Syntaxe (langage)
   - Sémantique (théorie des modèles)
   - Résolution
3. Logique des prédicats

Problématique

Vu jusqu'à maintenant : la structure du langage logique : Syntaxe (grec : syn-taxis : composition)
Dans cette section : la signification du langage logique : Sémantique (grec : sema : signe)

Dichotomie syntaxe / sémantique dans tout langage :

- en logique et dans les langages de programmation
- en linguistique (signifiant/signifié $\sim$ le mot/la chose)

Problématique

Vu jusqu'à maintenant : la structure du langage logique : Syntaxe (grec : syn-taxis : composition)
Dans cette section : la signification du langage logique : Sémantique (grec : sema : signe)

Dichotomie syntaxe / sémantique dans tout langage :

- en logique et dans les langages de programmation
- en linguistique (signifiant/signifié $\sim$ le mot/la chose)

On va maintenant précisément répondre à la question : quand est-ce qu'une formule est vraie ($= 1$) ou fausse ($= 0$) ?

Valuation

- En général, une formule propositionnelle n'est ni vraie ni fausse a priori. Cela dépend des valeurs associées aux variables de la formule.

Valuation

- En général, une formule propositionnelle n'est ni vraie ni fausse a priori. Cela dépend des valeurs associées aux variables de la formule.
- De la même façon, une équation (par exemple $x^2 + 1 = y$) n'est ni vraie ni fausse tant qu'on ne connaît pas les valeurs associées à x et à y, c'est-à-dire le contexte.
- Si le contexte associe à x la valeur 2, et à y la valeur 5, alors elle est vraie, dans d'autres contextes elle serait fausse.

Valuation

- En général, une formule propositionnelle n'est ni vraie ni fausse a priori. Cela dépend des valeurs associées aux variables de la formule.
- De la même façon, une équation (par exemple $x^2 + 1 = y$) n'est ni vraie ni fausse tant qu'on ne connaît pas les valeurs associées à x et à y, c'est-à-dire le contexte.
- Si le contexte associe à x la valeur 2, et à y la valeur 5, alors elle est vraie, dans d'autres contextes elle serait fausse.
- Réciproquement, on peut se demander quels sont les contextes qui la rendent vraies (= quelles en sont les solutions).

Valuation

- En général, une formule propositionnelle n'est ni vraie ni fausse a priori. Cela dépend des valeurs associées aux variables de la formule.
- De la même façon, une équation (par exemple $x^2 + 1 = y$) n'est ni vraie ni fausse tant qu'on ne connaît pas les valeurs associées à x et à y, c'est-à-dire le contexte.
- Si le contexte associe à x la valeur 2, et à y la valeur 5, alors elle est vraie, dans d'autres contextes elle serait fausse.
- Réciproquement, on peut se demander quels sont les contextes qui la rendent vraies (= quelles en sont les solutions).
- Les mêmes questions se posent quant à la vérité/fausseté des formules logiques...

Valuation

- Les contextes possibles pour une formule s'appelle des valuations
- Une valuation décrit quelle valeur de vérité est associée à chaque variable propositionnelle

Valuation

- Les contextes possibles pour une formule s'appelle des valuations
- Une valuation décrit quelle valeur de vérité est associée à chaque variable propositionnelle
- Rappel : l'ensemble de propositions est noté *PROP*.

Valuation

- Les contextes possibles pour une formule s'appelle des valuations
- Une valuation décrit quelle valeur de vérité est associée à chaque variable propositionnelle
- Rappel : l'ensemble de propositions est noté *PROP*.
- Formellement : une valuation est une fonction $v : PROP \Rightarrow \{0, 1\}$
- Arbitrairement on pose : $v(p) = 0$: "p faux" ; $v(p) = 1$: "p vrai" ;
- Étant donné une valuation, on peut, de proche en proche, calculer la valeur de vérité de n'importe quelle formule grâce aux tables de vérité (TdV) en calculant d'abord la valeur de vérité de ses sous-formules (voir aussi la définition d'une interprétation p. 75).

Tables de vérité

Table de vérité (TdV) des connecteurs

A	B	$A \wedge B$	$A \vee B$	$A \leftrightarrow B$	$A \oplus B$	$A \longrightarrow B$
0	0	0	0	1	0	?
0	1	0	1	0	1	?
1	0	0	1	0	1	?
1	1	1	1	1	0	?

La table de vérité recense toutes les valuations possibles.
Chaque ligne de la TdV correspond à une valuation.

Discussion

Quelle table de vérité pour $\longrightarrow$?

Tables de vérité

La TdV de l'implication : discussion

Dans un jeu, il y a des cartes avec une lettre sur une face et un nombre sur l'autre. À un certain moment du jeu, si un joueur pose une carte avec une voyelle sur une face, cette carte doit avoir un nombre pair sur l'autre face.
Vous voyez les cartes suivantes, lesquelles faut-il retourner pour vérifier que ce sont des cartes autorisées ?

$\boxed{A}$ $\boxed{Z}$ $\boxed{4}$ $\boxed{3}$

Tables de vérité

La TdV de l'implication : discussion

Dans un jeu, il y a des cartes avec une lettre sur une face et un nombre sur l'autre. À un certain moment du jeu, si un joueur pose une carte avec une voyelle sur une face, cette carte doit avoir un nombre pair sur l'autre face.

Vous voyez les cartes suivantes, lesquelles faut-il retourner pour vérifier que ce sont des cartes autorisées ?

$\boxed{A}$ $\boxed{Z}$ $\boxed{4}$ $\boxed{3}$

Avant de répondre, voici un deuxième exemple, identique mais plus concret.

Tables de vérité

La TdV de l'implication : discussion

Dans un bar, il est interdit de servir de l'alcool à un mineur (alcool$\longrightarrow$ majeur), cette loi est respectée dans un contexte donné si la formule en bleu est vraie : être majeur est une condition nécessaire pour y boire de l'alcool. Vous voyez les commandes suivantes, sur une face il y a la boisson commandée et sur l'autre l'âge du client.
Lesquelles faut-il retourner pour vérifier que ce sont des commandes autorisées ?

$\boxed{\text{Bière}}$ $\boxed{\text{Café}}$ $\boxed{\text{21 ans}}$ $\boxed{\text{13 ans}}$

Tables de vérité

La TdV de l'implication : discussion

Quels sont les contextes qui enfreignent la loi, ou, dit autrement, pour quelles valuations l'implication (alcool$\longrightarrow$ majeur) est-elle fausse ?

alcool	majeur	alcool $\longrightarrow$ majeur
0	0	?
0	1	?
1	0	?
1	1	?

Tables de vérité

La TdV de l'implication : discussion

Quels sont les contextes qui enfreignent la loi, ou, dit autrement, pour quelles valuations l'implication (alcool$\longrightarrow$ majeur) est-elle fausse ?

alcool	majeur	alcool $\longrightarrow$ majeur
0	0	?
0	1	?
1	0	?
1	1	?

On voit bien que si v(alcool)=0, la règle ne peut pas avoir été enfreinte et que si v(majeur)=1 non plus...
Le seul cas où elle l'est est quand v(alcool)=1 ET que v(majeur)=0 !

Tables de vérité

La TdV de l'implication : discussion (suite)

Autre exemple : soit f une fonction non dérivable, est-ce que le fait qu'elle ne soit pas dérivable rend fausse l'implication (f dérivable $\longrightarrow f$ continue) qui, on le sait, est vraie pour toute fonction ?

f-dérivable	f-continue	f-dérivable $\longrightarrow$ f-continue
0	0	1
0	1	1
1	0	0 (*)
1	1	1

Tables de vérité

La TdV de l'implication : discussion (suite)

Autre exemple : soit f une fonction non dérivable, est-ce que le fait qu'elle ne soit pas dérivable rend fausse l'implication (f dérivable $\longrightarrow$ f continue) qui, on le sait, est vraie pour toute fonction ?

f-dérivable	f-continue	f-dérivable $\longrightarrow$ f-continue
0	0	1
0	1	1
1	0	0 (*)
1	1	1

(*) seul contexte (valuation) qui pourrait démentir le théorème...
Prouver le théorème consiste à montrer que ce cas ne peut pas se produire.

Tables de vérité

TdV de l'implication (et de la négation)

A	B	$A \longrightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Tables de vérité

TdV de l'implication (et de la négation)

A	B	$A \longrightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

A	$\neg A$
0	1
1	0

Tables de vérité d'une formule

Table de vérité de $(p \wedge q) \vee (\neg r)$

p	q	r	$p \wedge q$	$\neg r$	$(p \wedge q) \vee (\neg r)$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

La table de vérité à 2^n lignes recense toutes les valuations possibles pour n variables propositionnelles.
Chaque ligne de la TdV correspond à une valuation.

Interprétation : définition formelle

Rappel : l'ensemble des formules est noté *FORM*.
À partir d'une valuation v on définit récursivement son extension v à toute formule de l'ensemble *FORM* de la manière suivante :

- $v(p) = v(p)$
- $v(\perp) = 0$
- $v(\top) = 1$
- $v(\neg A) = 1 - v(A)$
- $v(A \wedge B) = min(v(A), v(B))$
- $v(A \vee B) = max(v(A), v(B))$
- $v(A \longrightarrow B) = max(1 - v(A), v(B))$

Cette extension v est l'interprétation basée sur la valuation v.
On ne fera pas la distinction entre la valuation v et l'interprétation v.

Valuations et tables de vérité

Encore une fois la table de vérité de $(p \wedge q) \vee (\neg r)$

p	q	r	$p \wedge q$	$\neg r$	$(p \wedge q) \vee (\neg r)$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

Exemple : valuation v avec : $v(p) = 1, v(q) = 0, v(r) = 0$
$v((p \wedge q) \vee (\neg r)) = max(v(p \wedge q), v(\neg r)) =$
$max(min(v(p), v(q)), 1 - v(r)) = max(min(1, 0), 1 - 0) = max(0, 1) = 1$

Modèles

Modèle d'une formule : Une valuation v est

- un modèle d'une formule A si $v(A) = 1$
  On dit : v satisfait A
  Ex. : $v(p) = 0, v(q) = 1$ est un modèle de $p \vee q$
- un contre-modèle d'une formule A si $v(A) = 0$
  On dit : v falsifie A
  Ex. : $v(p) = 0, v(q) = 1$ est un contre-modèle de $p \wedge q$

- A est valide ssi pour tout v, $v(A) = 1$ (pas de contre-modèle)
- A est insatisfiable ssi pour tout v, $v(A) = 0$ (pas de modèle)
- A est satisfiable ssi il existe v, $v(A) = 1$ (au moins un modèle)

Valide/(In)satisfiable

<table>
<tr><td colspan="2">satisfiable</td><td>insatisfiable</td></tr>
<tr><td>⋮
$p \lor \neg p$
⋮</td><td>⋮
$p \land q$
⋮</td><td>⋮
$p \land \neg p$
⋮</td></tr>
<tr><td>valide</td><td colspan="2">non valide</td></tr>
</table>

Modèles

Formules valide, insatisfiable, satisfiable, leurs modèles et contre-modèles

p	q	r	$p \wedge q \longrightarrow q \vee r$	$(p \vee q) \wedge \neg p \wedge \neg q$	$(p \wedge q) \vee (\neg r)$
0	0	0	1	0	1
0	0	1	1	0	0
0	1	0	1	0	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	0	1

En bleu : modèles de la 3ème formule, en vert : ses contre-modèles. Toutes les valuations sont des modèles pour la 1ère, toutes sont des contre-modèles pour la 2ème.

Modèles

Modèle d'un ensemble de formules :

Soit H un ensemble de formules, une valuation v est

- un modèle de H si $v(A) = 1$ pour tout $A \in H$
  Ex. : $v(p) = 0, v(q) = 1$ est un modèle de $\{p \vee q, q\}$
- un contre-modèle de H si $v(A) = 0$ pour au moins un $A \in H$
  Ex. : $v(p) = 0, v(q) = 1$ est un contre-modèle de $\{p \vee q, p\}$

Exercices

Exercice 12 :

- Déterminez si ces formules sont valides / satisfiables / insatisfiables grâce à leurs TdV. Combien ont-elles de modèles ? De contre-modèles ? Donnez-en un exemple.
  1. $(p \vee q \longrightarrow r) \longrightarrow (p \longrightarrow r)$
  2. $(p \wedge q \longrightarrow r) \longrightarrow (p \longrightarrow r)$
  3. $\neg(p \vee (p \longrightarrow \perp)) \wedge q$
  4. $(p \wedge (q \longrightarrow r)) \longrightarrow ((\neg p \vee q) \longrightarrow (p \wedge r))$
- L'ensemble : $\{\neg p \longrightarrow m, m \longrightarrow g, \neg g\}$ est-il satisfiable ? Justifiez.

Exercice 13 : Rapports entre ces notions :

- Si A est valide, qu'est-ce que vous savez de $\neg A$?
- Si A est satisfiable, est-ce que $\neg A$ est insatisfiable ?

Conséquence (1)

Motivation : Définition rigoureuse et précise de ce que signifie tirer une conclusion à partir d'un ensemble d'hypothèses.

Exemple :

- H_1 : La fonction f est monotone
- H_2 : La fonction g est monotone
- C : La fonction $f \circ g$ est monotone

Notation : $\{H_1, H_2\} \models C$

On dit alors que C est une conséquence (logique) de H_1, H_2.

Définition

$H \models C$ ssi tout modèle de H est un modèle de C.

Cas particulier Ensemble d'hypothèses vide : $\{\} \models C$ ssi $\models C$ ssi C est valide

Conséquence (2)

p	q	r	$p \wedge q$	$\neg r$	$(p \wedge q) \vee (\neg r)$
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

Ici, on peut observer que $p, \neg r \models (p \wedge q) \vee (\neg r)$ car chaque fois que TOUTES les hypothèses sont vraies (en bleu), la conclusion l'est aussi (en vert). La conclusion est aussi vraie pour d'autres valuations, mais ce n'est pas la question.

Exercices

Exercice 14 :
Vrai ou faux ?

- $\{A, B\} \models A \vee B$
- $\{A, B\} \models A \wedge B$
- $\{A, B\} \models A \longrightarrow B$
- $\{A \longrightarrow B, B\} \models A$
- $\{A \longrightarrow B, A\} \models B$
- $\{A \longrightarrow B, \neg B\} \models \neg A$

Vérifiez

- si $\{A, B\} \models X$ alors $\{A, B, C\} \models X$
- $\{A, B\} \models X$ ssi $\{B, A\} \models X$
- $\{A, B\} \models X$ ssi $\{A\} \models B \longrightarrow X$
- $\{A, B\} \models \perp$ ssi $\{A\} \models \neg B$
- $\{A\} \models B$ ssi $\models A \longrightarrow B$

Équivalence

- Deux formules A et B sont équivalentes ssi elles ont les mêmes modèles. Elles sont également vraies ou fausses dans les mêmes contextes.

Équivalence

- Deux formules A et B sont équivalentes ssi elles ont les mêmes modèles. Elles sont également vraies ou fausses dans les mêmes contextes.
- Notation : $A \equiv B$

Équivalence

- Deux formules A et B sont équivalentes ssi elles ont les mêmes modèles. Elles sont également vraies ou fausses dans les mêmes contextes.
- Notation : $A \equiv B$
- Autrement dit, $A \equiv B$ ssi A et B ont les mêmes TdV.

Équivalence

- Deux formules A et B sont équivalentes ssi elles ont les mêmes modèles. Elles sont également vraies ou fausses dans les mêmes contextes.
- Notation : $A \equiv B$
- Autrement dit, $A \equiv B$ ssi A et B ont les mêmes TdV.

- Remarque 1 : $A \equiv B$ si et seulement si $A \leftrightarrow B$ est valide.
- Remarque 2 : $A \equiv B$ si et seulement si elles sont conséquences logiques l'une de l'autre ($A \models B$ et $B \models A$).

Équivalence

Exercice 15 : A et B sont des formules :

- Montrer que $A \wedge B \equiv B \wedge A$
- Montrer que $A \longrightarrow B \not\equiv B \longrightarrow A$ (déterminez A et B qui soient un contre-exemple)
- Montrer que $\neg A \equiv A \longrightarrow \bot$
- Montrer que $\neg(A \wedge B) \equiv \neg A \vee \neg B$

Équivalences remarquables

$\neg(A \vee B) \equiv \neg A \wedge \neg B$	$A \vee \top \equiv \top$	$A \vee \bot \equiv A$	$A \vee A \equiv A$
$\neg(A \wedge B) \equiv \neg A \vee \neg B$	$A \wedge \bot \equiv \bot$	$A \wedge \top \equiv A$	$A \wedge A \equiv A$
$A \vee B \equiv B \vee A$	$A \longrightarrow \top \equiv \top$	$\bot \longrightarrow A \equiv \top$	$A \vee \neg A \equiv \top$
$A \wedge B \equiv B \wedge A$	$A \longrightarrow \bot \equiv \neg A$	$\top \longrightarrow A \equiv A$	$A \wedge \neg A \equiv \bot$
$A \longrightarrow B \equiv \neg A \vee B$	$\neg\top \equiv \bot$	$\neg\neg A \equiv A$	

$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$	$A \wedge (B \vee C) \equiv A \wedge B \vee A \wedge C$
$(A \vee B) \vee C \equiv A \vee (B \vee C)$	$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

Équivalences

Exercice 16 :

1. En utilisant les équivalences ci-dessus, vérifiez la suivante : $(A \wedge B) \longrightarrow C \equiv A \longrightarrow (B \longrightarrow C)$
2. Exprimez le nouveau connecteur ! tel que la formule $!(A, B, C)$ est vraie si et seulement si exactement une formule sur les trois est vraie
3. Exprimez le nouveau connecteur !! tel que la formule $!!(A, B, C, D)$ est vraie si et seulement si exactement deux formules sur quatre sont vraies (Utilisez le connecteur !)
4. Exprimez les autres connecteurs uniquement à l'aide de $\neg$ et ! en fournissant des formules équivalentes

Théorème

Exercice 17 :

Démontrez que :

$H \models C$ si et seulement si $H \cup \{\neg C\}$ est insatisfiable

Résumé

- Différence syntaxe / sémantique
- On définit la sémantique d'une formule à l'aide d'une valuation étendue à une fonction d'interprétation
- Correspondance entre interprétations et tables de vérité
- Modèles d'une formule
- Validité, satisfiabilité
- Conséquence, équivalence logique, base pour le calcul booléen

Plan

1. Motivation
2. Logique des propositions
   - Syntaxe (langage)
   - Sémantique (théorie des modèles)
   - Résolution
3. Logique des prédicats

Motivation et Historique (1)

Conséquence logique : Une conséquence basée sur la sémantique (modèles des formules)
Dans cette section : Une autre notion de conséquence, purement syntaxique

Principes

- Travail formel sur la forme syntaxique des formules
- Déduction de nouvelles formules par application de règles : règles d'inférence

But

- Implantation simple et efficace sur ordinateur
- ...sans ambition d'être compréhensible pour des humains

Motivation et Historique (2)

Résolution : Une méthode de déduction assez récente :
A Machine-Oriented Logic Based on the Resolution Principle (Alan Robinson, 1965)

Contexte :

- Premières applications en Intelligence Artificielle

Principe :

- Une seule règle d'inférence appliquée à un ensemble de clauses
- …nécessitant un pré-traitement des formules du problème

Motivation et Historique (3)

Idée : Propagation de faits

Données :

- des faits : $p, \; r$
- des implications : $p \longrightarrow q, \; r \longrightarrow q \longrightarrow s, \; q \longrightarrow s \longrightarrow u$
- un but : prouver u

Démarche : Propagation des faits dans les implications ("modus ponens") :

1.
   - p et $p \longrightarrow q$ donnent q
   - r et $r \longrightarrow q \longrightarrow s$ donnent $q \longrightarrow s$
2.
   - q et $q \longrightarrow s \longrightarrow u$ donnent $s \longrightarrow u$
   - q et $q \longrightarrow s$ donnent s
3.
   - s et $s \longrightarrow u$ donnent u

Ceci est la base du langage de programmation Prolog

Motivation et Historique (4)

Les bases de la résolution :

- Au lieu d'implications ($p \longrightarrow q$) : des disjonctions : $\neg p \vee q$
- La règle de résolution : p et $\neg p \vee q$ donnent q

...dans un cadre plus général, avec plusieurs variables :

- Résolution :
  $p \vee q \vee \neg s$ et $\neg p \vee q \vee r$ donnent $q \vee r \vee \neg s$

Étapes principales : Pour une formule F

1. Construction d'une forme clausale de F [5]
2. Application de la résolution

5. On ne change pas la sémantique de F, la forme clausale est équivalente à F.

Formes Normales

Littéral

Un littéral est une variable propositionnelle ou sa négation,
par exemple : p, $\neg q$, mais pas : $p \vee \neg q$, ni $\perp$
p est le littéral complémentaire de $\neg p$ et réciproquement.

Forme Normale Conjonctive ou FNC

Une formule est en FNC si elle est une conjonction de disjonctions de littéraux.

Forme Normale Disjonctive ou FND

Une formule est en FND si elle est une disjonction de conjonctions de littéraux.

Formes Normales

Exercice 18 : FND, FNC, ou les deux ou aucune des deux ?

- $(\neg p \vee q) \wedge (r \vee s)$
- $(\neg p \wedge q) \vee (r \wedge s)$
- $\neg(p \wedge q) \vee (r \wedge s)$
- $p \vee \neg q$
- $(p \vee \bot) \wedge (r \wedge \neg s)$
- $p \vee (r \vee \neg s)$
- $p \wedge \neg q$
- p

Conversion en Forme Normale Conjonctive

1. Éliminer les connecteurs autres que $\neg, \wedge, \vee$
   1. Réécrire $p \leftrightarrow q$ en $(p \longrightarrow q) \wedge (q \longrightarrow p)$
   2. Réécrire $p \longrightarrow q$ en $\neg p \vee q$
2. Tirer les négations à l'intérieur, éliminer les doubles négations :
   - $\neg(p \wedge q)$ devient $\neg p \vee \neg q$
   - $\neg(p \vee q)$ devient $\neg p \wedge \neg q$
   - $\neg\neg p$ devient p
3. Distribuer $\vee$ sur $\wedge$:
   - $p \vee (q \wedge r)$ devient $(p \vee q) \wedge (p \vee r)$
   - et pareil pour $(q \wedge r) \vee p$
4. Eliminer $\perp$ et $\top$: $q \wedge (p \vee \perp)$, $p \vee \neg\perp \ldots$

On obtient une formule équivalente à la formule initiale (cf diapo 87)

Conversion en Forme Normale Conjonctive

Exercice 19 : Convertir en FNC :

- $\neg((p \wedge \neg q) \longrightarrow (p \leftrightarrow \neg q))$
- $\neg(((n \longrightarrow f) \wedge (v \longrightarrow f) \wedge (n \vee v) \wedge (e \longrightarrow \neg f)) \longrightarrow \neg f)$

Clauses

Clause

Une clause est une disjonction de littéraux. Elle peut être représentée par l'ensemble de ses littéraux.
Exemple : $\{p, \neg q\}$ représente la clause $p \vee \neg q$

Clauses

Clause

Une clause est une disjonction de littéraux. Elle peut être représentée par l'ensemble de ses littéraux.
Exemple : $\{p, \neg q\}$ représente la clause $p \vee \neg q$

Clause vide

La clause vide $\{\}$ représente $\perp$
Une disjonction de littéraux est satisfaite ssi l'un des littéraux est satisfait.
L'ensemble vide de littéraux est donc insatisfiable.

Clauses

Clause

Une clause est une disjonction de littéraux. Elle peut être représentée par l'ensemble de ses littéraux.
Exemple : $\{p, \neg q\}$ représente la clause $p \vee \neg q$

Clause vide

La clause vide $\{\}$ représente $\perp$
Une disjonction de littéraux est satisfaite ssi l'un des littéraux est satisfait.
L'ensemble vide de littéraux est donc insatisfiable.

Clause tautologique

Une clause contenant un littéral et son complémentaire est dite tautologique. Ainsi $\{p, q, \neg p\}$ représente $p \vee q \vee \neg p$, équivalente à $\top$.

NB : $\{p, \neg q\}$ représente $p \vee \neg q$ et $p \vee \neg q \vee p$ ($\neq$ mais équivalentes)

Forme clausale

- Une formule en FNC est une conjonction de clauses.
- Une conjonction de clauses peut être représentée par l'ensemble des clauses, donc un ensemble d'ensembles de littéraux.
  Exemple : $\{\{p, \neg q\}, \{r, \neg p\}\}$ représente $(p \vee \neg q) \wedge (r \vee \neg p)$
- La forme clausale d'une formule F est la conjonction de clauses associée à la forme normale conjonctive de F.
- F est équivalente à sa forme clausale.

Règle de Résolution

Résolvante

Soit deux clauses $\mathcal{C} = \{\ell_1 \dots \ell_m, p\}$ et $\mathcal{C}' = \{\ell'_1 \dots \ell'_n, \neg p\}$.
Leur résolution consiste à les fusionner en éliminant une seule paire de littéraux complémentaires, produisant ainsi
leur résolvante $\{\ell_1 \dots \ell_m, \ell'_1 \dots \ell'_n\}$

Cas particuliers

- La résolution des clauses $\{p\}$ et $\{\neg p\}$ produit la clause vide $\{\}$
- Pour les clauses $\{\neg p, q, r\}$ et $\{\neg q, p\}$, deux résolutions sont possibles, conduisant aux résolvantes $\{\neg q, q, r\}$ et $\{\neg p, p, r\}$, qui sont des clauses tautologiques, équivalentes à $\top$
- La résolvante des clauses $\mathcal{C}$ et $\mathcal{C}'$ étant unique à équivalence près, elle sera notée $res(\mathcal{C}, \mathcal{C}')$

Règle de Résolution

Exemples

- $res(\{p, q\}, \{\neg r, \neg q\}) : \{p, \neg r\}$
- $res(\{\neg p, q, r\}, \{\neg q, p\}) : \{\neg p, p, r\}$
- $res(\{\neg p, q, r\}, \{\neg q, p\}) : \{\neg q, q, r\}$

Les clauses représentées par $\{\neg p, p, r\}$ et $\{\neg q, q, r\}$ sont équivalentes (à $\top$)

Règle de Résolution

Ensemble des résolvantes

Soit $\mathcal{E}$ un ensemble de clauses, *ensres*$(\mathcal{E})$ dénote l'ensemble des résolvantes des clauses de $\mathcal{E}$

Exercice 20 : Calculez *ensres*$(\mathcal{E})$ où $\mathcal{E} = \{\{r\}, \{p, \neg q, r\}, \{\neg p, s\}, \{q, r, \neg s\}, \{\neg r\}, \{p, \neg s\}\}$
Signalez les clauses vides ou tautologiques.

Preuve d'insatisfiabilité par résolution

Les idées :

- Un ensemble de clauses $\mathcal{E}$ est saturé ssi il n'admet la dérivation d'aucune résolvante ($ensres(\mathcal{E}) \subseteq \mathcal{E}$)
- Etant donné $\mathcal{E}$ ensemble de clauses, $satur(\mathcal{E})$ dénote l'ensemble de clauses saturé obtenu par résolutions successives à partir de $\mathcal{E}$
- On prouve que $\mathcal{E}$ est insatisfiable ssi $\{\} \in satur(\mathcal{E})$

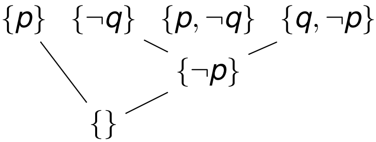
Preuve d'insatisfiabilité par résolution

Etant donné un ensemble de clauses $\mathcal{E}$, pour déterminer si $\mathcal{E}$ est insatisfiable :

Algorithme Res

`while` $\{\} \notin \mathcal{E}$ et $\mathit{ensres}(\mathcal{E}) \not\subseteq \mathcal{E}$:
    $\mathcal{E} = \mathcal{E} \cup \mathit{ensres}(\mathcal{E})$
    `if` $\{\} \in \mathcal{E}$:
        `print("insatisfiable")`
    `else:`
        `print("satisfiable")`

Exemple d'un ensemble de clauses insatisfiable :



Preuve d'insatisfiabilité par résolution

Les propriétés :
Etant donné $\mathcal{E}$ ensemble de clauses

1. $\mathcal{E}$ et $\mathit{ensres}(\mathcal{E})$ ont les mêmes modèles
2. Correction : Si l'algorithme Res produit $\{\}$, alors $\mathcal{E}$ n'est pas satisfiable
3. Complétude : Si l'algorithme Res produit $\mathit{satur}(\mathcal{E})$ avec $\{\} \notin \mathit{satur}(\mathcal{E})$, alors $\mathcal{E}$ est satisfiable

Résolution : Propriétés (1)

Préservation des interprétations

Soit v une valuation : $v(\mathcal{E} \cup ensres(\mathcal{E})\}) = v(\mathcal{E})$ pour $\mathcal{C}, \mathcal{C}' \in \mathcal{E}$

Preuve : Soit $\mathcal{C} = \{\ell_1 \dots \ell_m, p\}$, $\mathcal{C}' = \{\ell'_1 \dots \ell'_n, \neg p\}$

- Soit v un modèle de $\mathcal{E} \cup ensres(\mathcal{E})$. Alors, v est un modèle de $\mathcal{E}$.
- Soit v un modèle de $\mathcal{E}$, $\mathcal{C}, \mathcal{C}' \in \mathcal{E}$ avec $res(\mathcal{C}, \mathcal{C}') \in ensres(\mathcal{E})$
  alors $v(\mathcal{C}) = 1$ et $v(\mathcal{C}') = 1$
  donc $v(\ell_1 \vee \cdots \vee \ell_m \vee p) = 1$ et $v(\ell'_1 \vee \cdots \vee \ell'_n \vee \neg p) = 1$.
  Alors, $v(\ell_1 \vee \cdots \vee \ell_m \vee \ell'_1 \vee \cdots \vee \ell'_n) = 1$
  Pourquoi ? Complétez les détails !
  Donc v est un modèle de $\mathcal{E} \cup \{res(\mathcal{C}, \mathcal{C}')\}$.

Résolution : Propriétés (2)

Correction : Étant donné $\mathcal{E}$. Si l'algorithme de résolution Res fournit $\{\}$, alors $\mathcal{E}$ n'est pas satisfiable.

Preuve : par induction sur le nombre n d'itérations de l'algorithme.

- $n = 0$: Alors, $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k, \{\}\}$ représente la formule $C_1 \wedge \cdots \wedge C_k \wedge \bot \equiv \bot$
- hérédité de n à $n + 1$:
  D'après la préservation des interprétations :
  Si $\mathcal{E} \cup ensres(\mathcal{E})$ n'a pas de modèle, alors $\mathcal{E}$ non plus.

Résolution : Propriétés (3)

Complétude : Étant donné $\mathcal{E}$. Si l'algorithme de résolution Res fournit $satur(\mathcal{E})$ avec $\{\} \notin satur(\mathcal{E})$, alors $\mathcal{E}$ est satisfiable.
Preuve : par induction sur le nombre n d'itérations de l'algorithme.

- $n = 0$: Si
  - $\{\} \notin \mathcal{E}$
  - et $\mathcal{E}$ est saturé (n'admet pas la dérivation d'une nouvelle résolvante)

  alors $\mathcal{E}$ a un modèle.
  voir : Extraction d'un modèle
- hérédité de n à $n+1$:
  D'après la préservation des interprétations :
  Tout modèle de $\mathcal{E} \cup \{res(\mathcal{C}, \mathcal{C}')\}$ est un modèle de $\mathcal{E}$.

Extraction d'un modèle (1)

Construction d'un modèle de $\mathcal{E}$ dans le cas où $\mathcal{E}$ est saturé.

Les idées :

- Soit p une variable de $\mathcal{E}$. On considère deux ensembles simplifiés de clauses :
  - $\mathcal{E}[\bot/p]$: la variable p est supposée fausse
  - $\mathcal{E}[\top/p]$: la variable p est supposée vraie
- On prouve que si $\mathcal{E}$ est saturé, alors $\mathcal{E}[\bot/p]$ et $\mathcal{E}[\top/p]$ le sont aussi
- En itérant la construction,
  - si on trouve un ensemble vide de clauses, le chemin donne un modèle de $\mathcal{E}$.
  - si on trouve un ensemble de clauses contenant $\{\}$, le chemin donne un contre-modèle de $\mathcal{E}$.

Extraction d'un modèle (2)

Substitution d'une constante dans un ensemble de clauses

- $\mathcal{E}[\bot/p]$ est l'ensemble de clauses obtenu en :
  - éliminant le littéral p de toutes les clauses de $\mathcal{E}$
  - éliminant toute clause contenant $\neg p$

  Exemple : $\{\{p, \neg q\}, \{\neg p, r\}, \{p, r\}\}[\bot/p] = \{\{\neg q\}, \{r\}\}$
- $\mathcal{E}[\top/p]$ est l'ensemble de clauses obtenu en :
  - éliminant toute clause contenant p
  - éliminant le littéral $\neg p$ de toutes les clauses de $\mathcal{E}$

  Exemple : $\{\{p, \neg q\}, \{\neg p, r\}, \{p, r\}\}[\top/p] = \{\{r\}\}$

Justifiez cette définition

Montrez : Si $\mathcal{E}$ est saturé, alors $\mathcal{E}[\bot/p]$ et $\mathcal{E}[\top/p]$ aussi

Extraction d'un modèle (3)

Construction d'un modèle de $\mathcal{E}$ dans le cas où $\{\} \notin \mathcal{E}$ et $\mathcal{E}$ est saturé.

Induction sur le nombre n de variables $p_1, \dots p_n$ dans $\mathcal{E}$.
La preuve construit une valuation v de $p_1, \dots p_n$ telle que $v(\mathcal{E}) = 1$

- $n = 0$: $\mathcal{E}$ est de la forme :
  - $\{\}$: Alors, $\mathcal{E}$ est valide
  - $\{\{\}\}$: impossible, car par hypothèse $\{\} \notin \mathcal{E}$
- hérédité de n à $n+1$:
  1. Construire $\mathcal{E}' := \mathcal{E}[\bot/p_{n+1}]$, qui est saturé. Deux cas :
     - $\{\} \in \mathcal{E}'$: Alors, $\mathcal{E}'$ n'a pas de modèle.
     - $\{\} \notin \mathcal{E}'$: Par hypothèse d'induction, il existe v' tq $v'(\mathcal{E}') = 1$
       On définit v = $v'[p_{n+1} := 0]$, et on a $v(\mathcal{E}) = 1$
  2. Construire $\mathcal{E}'' := \mathcal{E}[\top/p_{n+1}]$, et procéder comme en (1)

  Montrez qu'on ne peut avoir $\{\} \in \mathcal{E}'$ et $\{\} \in \mathcal{E}''$.

Extraction d'un modèle (4)

Exemple : Ensemble de clauses initial : $\{\{p, \neg q\}, \{q, \neg r\}, \{p, r\}\}$

Ensemble saturé : $\mathcal{E} = \{\{p, \neg q\}, \{q, \neg r\}, \{p, r\}, \{p, \neg r\}, \{p, q\}, \{p\}\}$

Construction des modèles :

- $\mathcal{E}[\bot/p] = \{\{\neg q\}, \{q, \neg r\}, \{r\}, \{\neg r\}, \{\}\}$ n'a pas de modèle
- $\mathcal{E}[\top/p] = \{\{q, \neg r\}\}$
  - $\{\{q, \neg r\}\}[\bot/q] = \{\neg r\}$
    - $\{\neg r\}[\bot/r] = \{\}$ est valide
      $\rightsquigarrow$ valuation $v_0(p) = 1$, $v_0(q) = 0$, $v_0(r) = 0$
    - $\{\neg r\}[\top/r] = \{\{\}\}$ n'a pas de modèle
  - $\{\{q, \neg r\}\}[\top/q] = \{\}$ est valide
    $\rightsquigarrow$ valuation $v_1(p) = 1$, $v_1(q) = 1$

Optimisation

Subsomption : La clause $\mathcal{C}$ subsume la clause $\mathcal{C}'$ si $\mathcal{C} \subseteq \mathcal{C}'$.
Signification :
Si $\mathcal{C}$ et $\mathcal{C}'$ représentent les formules C et C', alors $C \models C'$.
Exemple : $\{p\}$ subsume $\{p, \neg q\}$, car $p \models (p \vee \neg q)$
Constat : Si $\mathcal{E} \cup \{\mathcal{C}, \mathcal{C}'\}$ est insatisfiable et $\mathcal{C} \subseteq \mathcal{C}'$, alors $\mathcal{E} \cup \{\mathcal{C}\}$ est insatisfiable
Optimisation : Pour la preuve de l'insatisfiabilité de $\mathcal{E}$, supprimer de $\mathcal{E}$ les clauses subsumées par d'autres.

Exercice 21 : Dérivez la clause vide à partir de $\mathcal{E} = \{\{p, \neg q\}, \{\neg p, r\}, \{p\}, \{q, r\}, \{\neg r\}\}$

Résolution : Résumé

Idée générale : La résolution permet de vérifier qu'un ensemble de clauses est insatisfiable.

Deux applications :

- Vérifier qu'une formule F est valide :
  F est valide ssi $\neg F$ est insatisfiable
- Vérifier que $\{H_1, \ldots H_N\} \models F$:
  $\{H_1, \ldots H_N\} \models F$ ssi $\{H_1, \ldots H_N, \neg F\}$ est insatisfiable

Résolution et preuve de validité

Pour vérifier la validité d'une formule F

1. Convertir $\neg F$ en Forme Normale Conjonctive
2. Construire l'ensemble de clauses correspondant
3. Appliquer l'algorithme de résolution :
   - Si $\{\}$ est une résolvante, alors F est valide
   - Si on ne peut pas obtenir $\{\}$, alors F n'est pas valide (contre-modèle obtenu par extraction d'un modèle de l'ensemble de clauses saturé)

Résolution et preuve de conséquence

Pour vérifier si $\{H_1, \dots H_N\} \models F$

- Convertir $(H_1 \wedge \cdots \wedge H_n \wedge \neg F)$ en FNC
- équivaut à convertir H_1 en FNC, $\dots H_n$ en FNC, $\neg F$ en FNC et prendre l'union des clauses
- Vérifier que cet ensemble est insatisfiable

Résolution

Exercice 22 : Vérifier si :

- $\{p \longrightarrow q, q \wedge r \longrightarrow \neg s, p, s\} \models \neg r$
- $\{n \longrightarrow f, v \longrightarrow f, n \vee v, e \longrightarrow \neg f\} \models \neg e$
- $(p \wedge (q \longrightarrow r)) \models ((\neg p \vee q) \longrightarrow (p \wedge r))$

Vérifier si ces ensembles de formules sont satisfiables (si oui donnez un modèle) :

- $\{a, b \longrightarrow c \wedge a, \neg b \longrightarrow \neg a\}$
- $\{p \vee q, p \leftrightarrow q, \neg(p \wedge q)\}$
- $\{2 \geq 1, 3 \geq 2, 3 \geq 1 \longrightarrow 1 \leq 3, \neg 1 \leq 3\}$

Résolution

Exercice 23 : Reprise de l'exercice 4 :

- Propositions : "PCM invasive", "PCM nécessitant une injection", "PCM nécessitant une sonde respiratoire", "Electrocardiogramme non normal", "Présence d'ondes P"
- Plus brièvement représentées par les variables : i, j, s, e, p, cela donne :
  1. i si et seulement si j ou s
  2. si (non j) alors si e alors s
  3. si p alors e
  4. donc, si (non i) alors (non p)
- Vérifiez par résolution que la conclusion EST conséquence logique des hypothèses.

Résolution

Solveurs SAT

- La majorité des solveurs SAT sont des implémentations efficaces de l'algorithme Res vu ci-dessus.
- Le logiciel TouIST que nous allons utiliser se charge de l'interface d'édition, de la mise en FNC (format DIMACS), et de la communication avec le solveur (envoi de la FNC, réception/affichage des modèles)

Plan

1. Motivation
2. Logique des propositions
3. Logique des prédicats
   - Syntaxe (langage)
   - Sémantique (théorie des modèles)

Plan

		Lo. prop.	Lo. prédicats	...
Syntaxe		✔	☞	
Sémantique		✔		
Modélisation		✔		
Méth. preuve	• Déd. nat.			
	• Tableaux			
	• Résolution	✔		
Outils	• TouIST	✔		

Motivation (1)

La logique des propositions est limitée en expressivité.
Comment modéliser “Si l'objet $o1$ est un mammifère, alors $o1$ est homéotherme (= à sang chaud)” :
Modélisons par : $M1 \longrightarrow H1$
Il faut une formule par objet, ainsi pour l'objet $o2$: $M2 \longrightarrow H2$. Y compris pour les objets qui ne sont pas des mammifères ! !
Même en utilisant les connecteurs généralisés :

$$\bigwedge_{o \in \mathcal{O}} M(o) \longrightarrow H(o)$$

Cela impliquerait que l'ensemble $\mathcal{O}$ des objets dont on peut parler :

- est fini et connu (impossible d'exprimer cela “en général”)
- et gigantesque si l'on veut être exhaustif

Motivation (2)

La logique des prédicats LPred raffine la logique des propositions.

Individus et prédicats

- Langage naturel : “Si l’objet $o1$ est bleu, alors $o1$ est aussi rond”
- LProp : $B(o1) \longrightarrow R(o1)$, où
  - $B(o1), R(o1)$ sont des énoncés élémentaires (on ne peut pas séparer $o1$ ou B de $B(o1)$)
    (on pourrait tout aussi bien écrire $p \longrightarrow q$)
- LPred : $B(o1) \longrightarrow R(o1)$, où
  - $o1$ désigne le sujet de la phrase : “l’objet $o1$”
  - B est le prédicat qui désigne la propriété : “est bleu”
  - c’est le même $o1$ dans les deux parties.
    (on pourrait tout aussi bien écrire $C(x) \longrightarrow C(x)$, l’objet désigné reste le même)

Motivation (3)

Quantificateurs

- Langage naturel : “Tout objet bleu est rond”
- LProp : Pas exprimable en général
- LPred : $\forall x.(B(x) \longrightarrow R(x))$

Opérations / Fonctions

- Langage naturel : “Toute mère est une femme”
- LProp : Pas exprimable en général
- LPred : $\forall x.Femme((Mere(x))$, où
  - x désigne un objet
  - *Mere* est une fonction qui associe un objet à sa mère
  - *Femme* est un prédicat [6]

6. en grammaire moderne, le précicat est “est une femme” dont le noyau est un verbe attributif)

Syntaxe de LPred : Termes et formules

Dans la logique propositionnelle, une expression syntaxique est une formule (interprétée par vraie ou fausse).

Syntaxe de LPred : Termes et formules

Dans la logique propositionnelle, une expression syntaxique est une formule (interprétée par vraie ou fausse).

Dans la logique des prédicats, nous avons deux catégories syntaxiques : les termes et les formules. Un terme représente un individu (ou une entité).

Syntaxe de LPred : Termes

Un terme peut représenter :

- un individu connu, identifié (ex. l'objet $o1$, *Tom*)
  par une constante
- un individu quelconque (ex. un objet quelconque, un étudiant)
  par une variable
- un individu créé (ex. l'objet $o1$ repeint, le père de Tom, la fille de Bob et Léa)
  au moyen d'une fonction

Syntaxe de LPred : Termes

Termes : Soit

- *VAR* un ensemble de variables (d'individus)
- FON_n un ensemble de fonctions n-aires (à n arguments)
  Les constantes sont les éléments de FON_0. On écrit c au lieu de $c()$.

L'ensemble *TERM* des termes est défini comme le plus petit ensemble tel que :

1. Variable : si $x \in VAR$, alors $x \in TERM$
2. Application de fonction : si $t_1 \in TERM, \ldots, t_n \in TERM$ et $f \in FON_n$, alors $f(t_1, \ldots, t_n) \in TERM$

Exemples : Soit $f \in FON_2$, $x, y \in VAR$, $g \in FON_1$, $\pi \in FON_0$

- $f(x, c) \in TERM$
- $f(x, g(f(y, \pi))) \in TERM$

Syntaxe de LPred : Termes

Exemple : Le monde de mes voisins
Comment décrire : Tom, Bob, Léa, le père de Tom, la fille de Bob et Léa ?

- Constantes : t, b, l
- Fonctions :
  $pereDe \in FON_1$
  $filleDe \in FON_2$
- Termes : $t, b, l, pereDe(t), filleDe(b, l)$

Syntaxe de LPred : Formules atomiques

Une formule atomique exprime une propriété qui met en relation des individus.

- la propriété est représentée par un prédicat
  ex. Sportif, Musicien, Bleu, PlusJeune
- les individus sont représentés par des termes

Soit $PRED_n$ un ensemble de prédicat n-aires (à n arguments) :
si $t_1 \in TERM, \ldots, t_n \in TERM$ et $P \in PRED_n$, alors $P(t_1, \ldots, t_n)$ est une formule atomique.

Exemple : Le monde de mes voisins
“La fille de Bob et Léa est musicienne” : $Musicien(filleDe(b, l))$
“Tom est plus jeune que Bob” : $PlusJeune(t, b)$

Syntaxe de LPred : Formules

Formules : Soit $PRED_n$ un ensemble des prédicats n-aires

L'ensemble $FORM$ des formules est défini comme le plus petit ensemble tel que :

1. Application de prédicat : si $t_1 \in TERM, \ldots, t_n \in TERM$ et $P \in PRED_n$, alors $P(t_1, \ldots, t_n) \in FORM$
2. Constante “faux” : $\perp \in FORM$
3. Négation : Si $A \in FORM$, alors $(\neg A) \in FORM$
4. Connecteurs binaires : Si $A \in FORM$ et $B \in FORM$,
   alors $(A \wedge B) \in FORM, (A \vee B) \in FORM, (A \longrightarrow B) \in FORM$
5. Quantificateur universel (“pour tout”) : Si $A \in FORM$ et $x \in VAR$, alors $(\forall x.A) \in FORM$
6. Quantificateur existentiel (“il existe”) : Si $A \in FORM$ et $x \in VAR$, alors $(\exists x.A) \in FORM$

Syntaxe de LPred : Formules

Exemple : Le monde de mes voisins

- "Il existe une personne musicienne et non sportive" :
  $(\exists x.(Musicien(x) \wedge (\neg Sportif(x))))$
- "Toute personne est plus jeune que son père" :
  $(\forall x.PlusJeune(x, pereDe(x)))$
- "Toute personne musicienne et sportive a un père musicien ou sportif" :
  $(\forall x.(Musicien(x) \wedge Sportif(x)) \longrightarrow$
  $(Musicien(pereDe(x)) \vee Sportif(pereDe(x))))$

Fonctions et Prédicats

Utilisation :

Les fonctions permettent de désigner des entités (objets) complexes, créés (obtenus) à partir d'autres objets :

- "une tarte aux pommes" : $tarte(pomme)$
- "le père de Tom" : $pereDe(Tom)$

Les prédicats permettent de décrire une situation, sous la forme d'une propriété qui met des objets en relation :

- "Eve achète une pomme" : $Acheter(eve, pomme)$
- "Eve mange une tarte aux pommes" : $Manger(eve, tarte(pomme))$
- "Tom est plus jeune que Bob" : $PlusJeune(t, b)$

Fonctions et Prédicats

Attention : Les arguments d'un prédicat sont des termes, pas des formules :
"Eve mange la pomme qu'elle a achetée" :

- Faux : $Manger(eve, Acheter(eve, pomme))$
- Correct : $Acheter(eve, pomme) \longrightarrow Manger(eve, pomme)$

Convention : nom de fonctions : minuscules ; de prédicats : majuscules

Limites de l'expressivité

Contournables : par exemple : propriétés temporelles :

- "Si le composant $c1$ tombe en panne, il émet éventuellement un signal"
- Peut être codé par :
  $\forall t.Panne(c1, t) \longrightarrow (\exists t'.t' > t \wedge Signal(c1, t'))$

Essentielles :

- Utilisation des fonctions comme objets :
  - Quantification sur des fonctions : "Toute fonction a une inverse" :
    Interdit : $\forall f.\exists g.\forall x. = (f(g(x)), x)$
  - Fonctions qui prennent des fonctions comme argument :
    Interdit : $\forall f.\forall x. = (f(inv(f, x)), x)$
- De même : Quantification sur des prédicats ou ensembles

$\rightsquigarrow$ inapproprié pour raisonner sur certains programmes fonctionnels

Modélisation

Exercice 24 : Dans une soirée mondaine, autour d'une table circulaire, il y des hommes et des femmes, blond(e)s, brun(e)s ou roux/rousses, des grand(e)s, des moyen(ne)s et des petit(e)s. D'où vous êtes vous constatez les faits suivants :

1. Il n'y a aucun roux et aucune rousse.
2. Aucune personne blonde n'est brune et vice-versa.
3. Toute femme a un homme grand à sa droite.
4. Tout homme a une femme brune à sa gauche ou une femme blonde à sa droite.
5. Toute personne a sur sa droite une personne ayant sur sa droite une personne blonde.
6. Toute personne a sur sa droite une personne qui n'a que des blond(e)s sur sa gauche.
7. Il y a toujours une femme brune entre deux hommes petits quelconques.
8. Il n'y a pas deux personnes blondes de la même taille.

Formalisez ces énoncés en utilisant les prédicats suivants :

- $Bl(x)$: x est blond(e)
- $H(x)$: x est un homme
- $P(x)$: x est petit
- $x = y$: x et y sont identiques
- $Rx(x)$: x est roux/rousse
- $F(x)$: x est une femme
- $M(x)$: x est moyen
- $Br(x)$: x est brun(e)
- $Gd(x)$: x est grand
- $Dte(x, y)$: y est à droite de x

NB : Deux personnes sont de la même taille ssi elles sont toutes deux grandes ou petites ou moyennes ; Une personne en a une autre à sa gauche ssi la 2ème a la première à sa droite ; Une personne est entre deux autres ssi elle est à la droite de l'une et à gauche de l'autre.

Modélisation

Exercice 25 : En utilisant les prédicats et fonctions suivants :

- $Ent(x)$ se lit : x est un nombre entier ;
- $Rat(x)$ se lit : x est un nombre rationnel ;
- $Pair(x)$ se lit : l'entier x est pair ;
- $Pre(x)$ se lit : l'entier x est premier ;

- $y|x$ se lit : l'entier y divise l'entier x ;
- $S(x, y)$ se lit : l'entier y est le successeur de l'entier x (donc $y = x + 1$) ;
- $P(x, y)$ se lit : l'entier y est le prédécesseur de l'entier x (donc $y = x - 1$) ;
- les prédicats $=, <, \leq$, et les fonctions $0, 1, +$ et $*$ avec leur signification habituelle

Formalisez les énoncés :

1. Il y a un entier plus petit ou égal à tous les autres.
2. Il n'y a pas d'entier supérieur ou égal aux autres autres mais pour tout entier il y en a un plus grand.
3. Tout nombre entier pair est égal à la somme de deux nombres entiers premiers.

Définissez les notions suivantes :

4. Donnez une formule définissant $y|x$ par rapport aux autres prédicats et fonctions .
5. De même pour *Pair*, *Pre*, *P*, par rapport aux autres prédicats et fonctions
6. Définissez $Rac(x, y)$ où y est la partie entière de la racine carrée de x
7. Définissez $PGCD(x, y, z)$ où z est le pgcd de x et de y

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x. \forall y. P(f(x), z) \wedge Q(y)) \vee \forall x. R(x, z, g(x))$

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x.\, \forall y.\, P(f(x), z) \wedge Q(y)) \vee \forall x.\, R(x, z, g(x))$
  Liées ?

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x. \forall y. P(f(x), z) \wedge Q(y)) \vee \forall x. R(x, z, g(x))$
  Liées ?

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x. \forall y. P(f(x), z) \wedge Q(y)) \vee \forall x. R(x, z, g(x))$
  Liées ? Libres ?

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x.\, \forall y.\, P(f(x), z) \wedge Q(y)) \vee \forall x.\, R(x, z, g(x))$
  Liées ? Libres ?

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x.\, \forall y.\, P(f(x), z) \wedge Q(y)) \vee \forall x.\, R(x, z, g(x))$
  Liées ? Libres ? Liantes ?

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x. \forall y. P(f(x), z) \wedge Q(y)) \vee \forall x. R(x, z, g(x))$
  Liées ? Libres ? Liantes ?
- Une formule sans occurrence libre de variables est close.

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x.\, \forall y.\, P(f(x), z) \wedge Q(y)) \vee \forall x.\, R(x, z, g(x))$
  Liées ? Libres ? Liantes ?
- Une formule sans occurrence libre de variables est close.

Exercice : écrire la définition formelle !

Variables libres et variables liées

- Toute occurrence d'une variable dans une formule est liée ou libre ou liante.
- Exemple :
  $(Q(x) \vee \exists x. \forall y. P(f(x), z) \wedge Q(y)) \vee \forall x. R(x, z, g(x))$
  Liées ? Libres ? Liantes ?
- Une formule sans occurrence libre de variables est close.

Exercice : écrire la définition formelle !
Nous écrivons : $fv(F)$ pour les variables libres dans F et $v(s)$ pour les variables d'un terme s.

Substitutions (1)

- Substitution dans un terme :
  $t[s/x]$, où x est une variable et t et s sont des termes :
  1. Variable : $x[s/x] = s$ et $y[s/x] = y$ pour $y \neq x$
  2. Application de fonction : $f(t_1, \ldots, t_n)[s/x] = f(t_1[s/x], \ldots, t_n[s/x])$
- Substitution dans une formule :
  $F[s/x]$, où x est une variable, s est un terme et F une formule :
  1. Application de prédicat : $P(t_1, \ldots, t_n)[s/x] = P(t_1[s/x], \ldots, t_n[s/x])$
  2. Const. “faux”, connecteurs propositionnels : comme d’habitude
  3. Quantificateur universel :
     1. $(\forall x.A)[s/x] = (\forall x.A)$
     2. $(\forall y.A)[s/x] = (\forall y.(A[s/x]))$ pour $y \neq x$ avec $y \notin v(s)$
     3. $(\forall y.A)[s/x] = (\forall y'.(A[y'/y][s/x]))$ pour $y \neq x$ avec $y \in v(s)$, où y' est une variable “fraîche” (c.-à-d. $y' \notin v(s), y' \notin fv(A)$)
  4. Quantificateur existentiel : En analogie avec $\forall$

Substitutions (2)

Le renommage dans le troisième cas des quantificateurs assure qu'une substitution est saine : la variable quantifiée y diffère des variables de s.

Exemple : $(\forall y.R(y, x))[f(y)/x]$ =

- Faux : $(\forall y.R(y, f(y)))$, car l'occurrence libre dans $f(y)$ est capturée par accident.
- Correct : $(\forall z.R(z, f(y)))$

Substitutions (2)

Le renommage dans le troisième cas des quantificateurs assure qu'une substitution est saine : la variable quantifiée y diffère des variables de s.

Exemple : $(\forall y.R(y,x))[f(y)/x] =$

- Faux : $(\forall y.R(y,f(y)))$, car l'occurrence libre dans $f(y)$ est capturée par accident.
- Correct : $(\forall z.R(z,f(y)))$

Exercice : Calculer les substitutions

- $(\forall x.\exists y.R(x,y) \wedge P(z))[f(v)/z]$
- $(\forall x.\exists y.R(x,y) \wedge P(z))[f(x)/z]$
- $(\forall x.\exists y.R(x,y) \wedge P(z))[f(v)/x]$

Plan

1. Motivation
2. Logique des propositions
3. Logique des prédicats
   - Syntaxe (langage)
   - Sémantique (théorie des modèles)

Plan

		Lo. prop.	Lo. prédicats	...
Syntaxe		✔	✔	
Sémantique		✔	☞	
Modélisation		✔	☞	
Méth. preuve	• Déd. nat.	✔		
	• Tableaux	✔		
	• Résolution			
Outils	• TouIST	✔		

Valuation et Interprétation

Rappel : Logique propositionnelle

- Une valuation v détermine la valeur de vérité d'une variable propositionnelle. Exemple : $v(A) = 0$, $v(B) = 1$
- Une interprétation v étend une valuation à une formule. Exemple : $v(\neg A \wedge B) = 1$

Logique des prédicats : Plus complexe :

- Distinction entre “termes” et “formules” :
  $\rightsquigarrow$ deux fonctions d'interprétation différentes

Interprétation des termes (1)

Premier exemple intuitif : Quelle est la signification du terme $f(x, y)$?

- Si x représente 2, y représente 3 et f représente la fonction d'addition, alors $f(x, y)$ représente $2 + 3$, donc 5.
- Si x représente 2, y représente 3 et f représente la fonction de multiplication, alors $f(x, y)$ représente $2 * 3$, donc 6.

Interprétation des termes (1)

Premier exemple intuitif : Quelle est la signification du terme $f(x, y)$?

- Si x représente 2, y représente 3 et f représente la fonction d'addition, alors $f(x, y)$ représente $2 + 3$, donc 5.
- Si x représente 2, y représente 3 et f représente la fonction de multiplication, alors $f(x, y)$ représente $2 * 3$, donc 6.

Une valuation pour les termes est donnée par :
$(D, v_{VAR}, \{v_{FON_n} \mid n \in \mathbb{N}\})$ avec

- Un domaine d'interprétation D : un ensemble non vide
- v_{VAR} : $VAR \Rightarrow D$ est une valuation pour les variables
- v_{FON_n} : $FON_n \Rightarrow (D^n \Rightarrow D)$ est une valuation pour les fonctions

Interprétation des termes (2)

Exemple :

- Domaine $D = \mathbb{N}$
- Interprétation des variables : $v_{VAR} : VAR \Rightarrow \mathbb{N}$ avec $v_{VAR}(x) = 2$, $v_{VAR}(y) = 3$
- Interprétation des fonctions binaires :
  $v_{FON_2}(f)$ = la fonction d'addition
  $v_{FON_2}(g)$ = la fonction de multiplication

Le terme $f(x, g(x, y))$ est interprété par $2 + (2 * 3) = 8$

Interprétation des termes (3)

Une interprétation des termes v est l'extension de $(D, v_{VAR}, \{v_{FON_n} | n \in \mathbb{N}\})$ à *TERM* :

- Variable : $v\ (x) = v_{VAR}(x)$
- Application de fonction :
  $v\ (f(t_1, \ldots, t_n)) = v_{FON_n}(f)(v(t_1),\ \ldots,\ v(t_n))$

Exemple :

$v = (\mathbb{N}, v_{VAR} = [x \mapsto 2, y \mapsto 3], \{v_{FON_2} = [f \mapsto (+),\ g \mapsto (*)]\})$

$v\ (f(x, g(x, y))) = 2 + 6 = 8$:

- $v_{FON_2}(f) = (+)$
- $v\ (x) = v_{VAR}(x) = 2$
- $v\ (g(x, y)) = 2 * 3 = 6$
  - $v_{FON_2}(g) = (*)$
  - $v\ (x) = v_{VAR}(x) = 2$
  - $v\ (y) = v_{VAR}(y) = 3$

Interprétation des formules (1)

Quelle est la signification de la formule (élémentaire) $P(x)$?

1. Si x représente 2 et P représente la propriété “être un nombre pair”, alors $P(x)$ est une proposition vraie.
2. Si x représente π et P représente la propriété “être un nombre rationnel”, alors $P(x)$ est une proposition fausse.

Une valuation pour les formules est donnée par $(D, v_{VAR}, \{v_{FON_n} | n \in \mathbb{N}\}, \{v_{PRED_n} | n \in \mathbb{N}\})$:

- $(D, v_{VAR}, \{v_{FON_n} | n \in \mathbb{N}\})$ est une valuation pour les termes
- $v_{PRED_n} : PRED_n \Rightarrow (D^n \Rightarrow \{\mathbf{0}, \mathbf{1}\})$ est une valuation pour les prédicats

Interprétation des formules (2)

Une interprétation des formules v est l'extension de $v = (D, v_{VAR}, \{v_{FON_n} | n \in \mathbb{N}\}, \{v_{PRED_n} | n \in \mathbb{N}\})$ à $FORM$:

- Application de prédicat :
  $v(P(t_1, \ldots, t_n)) = v_{PRED_n}(P)(v(t_1), \ \ldots, \ v(t_n))$
- Constante "faux", négation, connecteurs binaires :
  comme pour la logique propositionnelle
- Quantificateur universel :
  $v(\forall x.\phi) = \begin{cases} 1 & \text{ssi } v_{[x:=d]}(\phi) = 1 \text{ pour tout } d \in D \\ 0 & \text{autrement} \end{cases}$
- Quantificateur existentiel :
  $v(\exists x.\phi) = \begin{cases} 1 & \text{ssi } v_{[x:=d]}(\phi) = 1 \text{ pour quelque } d \in D \\ 0 & \text{autrement} \end{cases}$

Restriction d'une valuation : $v_{[x:=d]}$ ne diffère de v que par v_{VAR}

$$v_{VAR_{[x:=d]}}(y) = \begin{cases} d & \text{si } x = y \\ v(y) & \text{si } x \neq y \end{cases}$$

Interprétation des formules (3)

Cas spécial : Domaine fini.

- $N_3 = \{0, 1, 2\}$
- $\oplus$ l'addition modulo 3 : $x \oplus y \equiv (x + y) \mod 3$

Soit la valuation v telle que :

- Domaine N_3
- $v_{VAR} = [x_1 \mapsto 1, x_2 \mapsto 2]$
- $\{v_{FON_0} = [n_0 \mapsto 0, n_1 \mapsto 1, n_2 \mapsto 2], v_{FON_2} = [a \mapsto \oplus]\})$
- $\{v_{PRED_1} = [P \mapsto$ "est un nombre pair"$]\}$

Interprétation de :

- $v(P(x_1)) = v_{PRED_1}(P)(v_{VAR}(x_1)) = pair(1) = \mathbf{0}$
- $v(\exists x.P(a(x_2, x)))$ = **1** car $v_{[x:=0]}(P(a(x_2, x))) = pair(2 \oplus 0) = \mathbf{1}$

Interprétation des formules (4)

Exercice 26 : On considère le domaine N_3 et la valuation v associée

- Développez $v(\forall x.P(a(x_2, x)))$
- Comparez $v(\exists x.P(x))$ et $v(P(n_0) \vee P(n_1) \vee P(n_2))$
- Comparez $v(\forall x.P(x))$ et $v(P(n_0) \wedge P(n_1) \wedge P(n_2))$

Interprétation des formules (4)

Les notions de modèle, satisfiabilité, validité, conséquence et équivalence logique sont définies comme en logique propositionnelle.

Exercice 27 : Dans le cas général

- Montrez : $\exists x.(P(x) \vee Q(x)) \equiv (\exists x.P(x)) \vee (\exists x.Q(x))$
- Montrez : $\forall x.(P(x) \wedge Q(x)) \equiv (\forall x.P(x)) \wedge (\forall x.Q(x))$
- Vous savez que $\neg(F_0 \vee F_1) \equiv (\neg F_0 \wedge \neg F_1)$.
  Alors : $\neg(\exists x.P(x)) \equiv ???$
- Vous savez que $\neg(F_0 \wedge F_1) \equiv (\neg F_0 \vee \neg F_1)$.
  Alors : $\neg(\forall x.P(x)) \equiv ???$

Interprétation des formules (4)

Exercice 28 : Vrai ou faux ?

- Si vrai, donnez un modèle fini (avec $\wedge$ et $\vee$)
  (c'est un argument de plausibilité, pas une preuve) !
- Si faux, donnez un contre-modèle

$\forall x.\forall y.R(x,y)$ $\equiv \forall y.\forall x.R(x,y)$	vrai	$(R(1,1) \wedge R(1,2)) \wedge (R(2,1) \wedge R(2,2))$ $\equiv (R(1,1) \wedge R(2,1)) \wedge (R(1,2) \wedge R(2,2))$
$\forall x.\exists y.R(x,y)$ $\equiv \exists y.\forall x.R(x,y)$	faux	contre-mod. : domaine : $\mathbb{N}$, $v_{PRED}(R) = <$
$\exists x.\exists y.R(x,y)$ $\equiv \exists y.\exists x.R(x,y)$		
$\exists x.(P(x) \wedge Q(x))$ $\equiv (\exists x.P(x)) \wedge (\exists x.Q(x))$		
$\forall x.(P(x) \vee Q(x))$ $\equiv (\forall x.P(x)) \vee (\forall x.Q(x))$		

Logique des prédicats : Résumé

- La logique des prédicats est beaucoup plus expressive que le logique propositionnelle :
  - Elle permet de distinguer les individus et les relations entre individus ;
  - Elle permet de parler d'une infinité d'individus ;
  - Elle permet des expressions plus concises . . .

Logique des prédicats : Résumé

- La logique des prédicats est beaucoup plus expressive que le logique propositionnelle :
  - Elle permet de distinguer les individus et les relations entre individus ;
  - Elle permet de parler d'une infinité d'individus ;
  - Elle permet des expressions plus concises . . .
- On a maintenant deux catégories syntaxiques : les termes et les formules.
- On a les quantificateurs $\forall, \exists$

Logique des prédicats : Résumé

- La logique des prédicats est beaucoup plus expressive que le logique propositionnelle :
  - Elle permet de distinguer les individus et les relations entre individus ;
  - Elle permet de parler d'une infinité d'individus ;
  - Elle permet des expressions plus concises . . .
- On a maintenant deux catégories syntaxiques : les termes et les formules.
- On a les quantificateurs $\forall, \exists$
- La logique des prédicats est la logique “par excellence” :
  - Il y a des logiques plus expressives, mais elles sont beaucoup moins connues ;
  - Il y a d'autres logiques (modale, temporelle, descriptive, . . .) qui peuvent être traduites dans la logique des prédicats.