

Introduction à la Logique et aux preuves

Didier Buchs

Université de Genève





16 novembre 2015

Plan

- Logique propositionnelle
- Logique des prédicats

Quels sont les buts de la logique ?

- Comprendre la nature intime du **raisonnement mathématique**¹
- Faire du “raisonnement” une **théorie mathématique** comme les autres.
- Donner un sens précis à ce que peut-être le vrai
dès qu’il s’agit de raisonnement et d’argumentation.

1. et du raisonnement non mathématique (philosophique, judiciaire)    

La logique propositionnelle

Sujets abordés :

- Syntaxe
- Sémantique : interprétation, tables de vérité, satisfaction
- Formes normales
- Théorie de la preuve :
 - Systèmes de Hilbert
 - Dédution Naturelle de Gentzen
 - Calcul des Séquents

La logique propositionnelle :Syntaxe

- Les symboles propositionnels P, Q, R, \dots et les constantes t et f
- Les formules sont construites à partir de : $\wedge, \vee, \neg, \rightarrow$
- Les précédences permettent d'omettre des parenthèses

Soit AP les variables propositionnelles, $F(AP)$ est définis comme le plus petit ensemble tel que :

- si $AP \subseteq F(AP)$
- si $g, h \in F(AP)$ alors $h \wedge g \in F(AP)$
- si $g, h \in F(AP)$ alors $h \vee g \in F(AP)$
- si $g, h \in F(AP)$ alors $h \rightarrow g \in F(AP)$
- si $g, h \in F(AP)$ alors $h \leftrightarrow g \in F(AP)$
- $f \in F(AP)$ et $t \in F(AP)$
- si $g \in F(AP)$ alors $\neg g \in F(AP)$

La logique propositionnelle :Sémantique(1/2)

- Une interprétation est une fonction des variables propositionnelles dans $\{t, f\}$
- Une interprétation satisfait une formule si elle est évaluée à t
- Une formule F est valide si elle est satisfaite pour toutes interprétations
- Une formule F est insatisfiable si elle n'est pas satisfiable
- Un ensemble de formule S est satisfiable si toutes ses formules sont satisfiables
- Un ensemble de formule S est insatisfiable si aucune de ses formules est satisfiable
- Un ensemble de formule S 'entails' (entraîne) A si pour chaque interpretation satisfaisant S , A est également satisfait.
 $S \models A$
- Si $A \models B$ et $B \models A$ alors $A \equiv B$

La logique propositionnelle :Sémantique(2/2)

- Chaque formule à un sens $\in \{t, f\}$ dépendant de la valeur des variables propositionnelles (leurs interprétations)
- Les tables de vérités déterminent la sémantique des opérateurs :

p	q	$p \vee q$	$p \wedge q$	$p \Rightarrow q$	$\neg p$
f	f	f	f	t	t
f	t	t	f	t	t
t	f	t	f	f	f
t	t	t	t	t	f

La logique propositionnelle : Calcul de la Sémantique

Exemple : $\neg P \wedge Q \vee R \rightarrow \neg P \vee Q$

La logique propositionnelle : Equivalences

- Différentes lois sont vérifiées :
- Idempotences : $A \wedge A \equiv A$, $A \vee A \equiv A$
- Commutativité : $A \wedge B \equiv B \wedge A$, $A \vee B \equiv B \vee A$
- Associativité : $(A \wedge B) \wedge C \equiv B \wedge (A \wedge C)$,
 $(A \vee B) \vee C \equiv B \vee (A \vee C)$
- Distributivité : $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$,
 $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
- Lois de de Morgan : $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$,
 $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$
- Définitions de connecteurs : $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$,
 $\neg A \equiv (A \rightarrow f)$, $A \rightarrow B \equiv (\neg A \vee B)$
- règles de négation : $\neg(A \rightarrow B) \equiv (A \wedge \neg B)$,
 $\neg(A \leftrightarrow B) \equiv (A \leftrightarrow \neg B) \equiv (\neg A \leftrightarrow B)$
- règles de simplification $A \wedge f \equiv f$, $A \wedge t \equiv A$, $A \vee f \equiv A$,
 $A \vee t \equiv t$, $\neg\neg A \equiv A$, $A \wedge \neg A \equiv f$, $A \vee \neg A \equiv t$,

La logique propositionnelle : Formes normales

- Redondance du langage ce qui implique de nombreuses formes équivalentes
- Réduction du nombre de connecteurs dans la formes normale
- Une forme normale peut -être exponentiellement plus grande

La logique propositionnelle : Définition Formes normales

- Un **litéral** : c'est une formule atomique ou sa négation
- Un **maxterm** est un litéral ou une disjonction de litéraux
- Un **minterm** est un litéral ou une conjonction de litéraux
- Une formule est sous forme **NNF** (Negation Normal Form) si les seuls connecteurs sont \wedge , \vee et \neg , et \neg est seulement appliqué aux litéraux.
- Une formule est sous forme **CNF** (Conjunctive Normal Form) si elle est de la forme $A_1 \wedge A_2 \wedge \dots \wedge A_m$ ou chaque A_i est un maxterm.
- Une formule est sous forme **DNF** (Disjunctive Normal Form) si elle est de la forme $A_1 \vee A_2 \vee \dots \vee A_m$ ou chaque A_i est un minterm.

La logique propositionnelle : Traduction en Formes Normales

La traduction se fait en plusieurs étapes

- **étape 1** : Eliminer les connecteurs implication et équivalence :

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A),$$

$$A \rightarrow B \equiv (\neg A \vee B)$$

- **étape 2** : Introduire les négations à l'intérieur des formules :

$$\neg \neg A \equiv A$$

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

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

- **étape 3** : Pour obtenir **CNF** mettre les disjonctions à l'intérieur en utilisant : $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
 $(B \wedge C) \vee A \equiv (B \vee A) \wedge (C \vee A)$

- **étape 4** : utiliser pour simplifier les règles :

- Supprimer les maxterms avec P et $\neg P$.

- Supprimer les maxterms contenant un autre maxterm :

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

La logique propositionnelle : Traduction en Formes Normales

Si on veut traduire en DNF

- étape 1 et étape 2 idem que précédemment pour CNF
- étape 3' et étape 4' : Utiliser les autres lois de distributions

La logique propositionnelle : Traduction en Formes Normales

Exemple : $P \vee Q \rightarrow Q \vee R$

La logique propositionnelle : Vérification de tautologies utilisant CNF

Il faut trouver une interpretation falsifiant la formule

La logique propositionnelle : Traduction en Formes Normales

Vérification de tautologies utilisant CNF Exercice : $P \wedge Q \rightarrow Q \wedge P$

La logique propositionnelle : Traduction en Formes Normales

Exercice : $((P \rightarrow Q) \rightarrow P) \rightarrow P$

Théorie de la preuve

- Notion de jugement
- Dédutions
- Systèmes de déduction :
 - Hilbert
 - Dédution naturelle
 - Calcul des séquents

Les jugements

Le concept de base de la théorie de la démonstration est le **jugement**.

Un jugement s'écrit $\Gamma \vdash \Delta$ et est constitué de regroupement de propositions.

- Si Γ est vide et Δ ne contient qu'une proposition, c'est l'**approche à la Hilbert**.
- Quand Γ est un multienemble (une structure de données ou l'ordre ne compte pas, mais ou les éléments peuvent être répétés) et Δ ne contient qu'une proposition, on a affaire à la **déduction naturelle**.

Les jugements (2/2)

Cas généraux :

- Quand Γ et Δ sont des multiensembles de propositions, on parle de **calcul des séquents**.
- Si Γ et Δ sont des multi-ensembles, mais si l'on contrôle très strictement l'emploi des duplications dans les preuves, – une proposition ne sert qu'une fois dans chaque preuve – on parle de **logique linéaire**.
- Si les propositions déclarent le type d'un élément, on parle de **jugements de typage**.

Déductions

$$\begin{array}{c}
 \frac{\frac{(\vdash \neg\phi \vee \neg\psi) \quad \frac{\frac{\vdash \phi \quad \overline{\vdash \neg\phi}}{\perp}}{\perp}}{\vdash \neg(\phi \wedge \psi)} \quad (1) \quad (2)} \quad (3)
 \end{array}$$

$$\frac{\text{premise}_1 \quad \dots \quad \text{premise}_n}{\text{conclusion}}$$

$$\frac{\frac{\text{premise}_{1,1} \quad \dots \quad \text{premise}_{1,n_1}}{\text{conclusion}} \quad \dots \quad \frac{\text{premise}_{n,1} \quad \dots \quad \text{premise}_{n,n_n}}{\text{conclusion}_n}}{\text{conclusion}}$$

logique propositionnelle minimale : La syntaxe 1/2

Il n'y a qu'un **connecteur** \rightarrow

et des **variables propositionnelles** $p, q, \dots, p_1, p_2, \dots$

Par exemple, les propositions sont

- p ,
- $p \rightarrow q$,
- $(p \rightarrow q) \rightarrow p$.

La syntaxe 2/2

On adopte la convention d'**associativité à droite** à savoir que

$$p_1 \rightarrow (p_2 \rightarrow \dots \rightarrow (p_{n-1} \rightarrow p_n) \dots)$$

s'écrit

$$p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_{n-1} \rightarrow p_n$$

Les jugements de logique à la Hilbert

Les jugements sont de la forme $\vdash \varphi$ ou φ est une proposition.
Ainsi on **distingue** certaines propositions des autres.

Question : Quelles sont en logique les propositions que l'on veut distinguer des autres ?

\Rightarrow tautologies obtenues par dérivation depuis des axiomes

Connecteurs

Ici seulement \neg et \rightarrow seront considérés. Les autres connecteurs seront définis de la manière suivante :

- $T \equiv p \vee \neg p$
- $F \equiv p \wedge \neg p$
- $A \wedge B \equiv \neg(A \rightarrow \neg B)$
- $A \vee B \equiv \neg A \rightarrow B$
- $A \leftrightarrow B \equiv (B \rightarrow A) \wedge (A \rightarrow B)$

Règles du système de preuve

Les constituants de la logique propositionnelle à la Hilbert. C'est une logique de presque rien :

- des jugements rudimentaires,
- trois axiomes.

$\vdash A \rightarrow (B \rightarrow A)$ axiome K

$\vdash (A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow B) \rightarrow (A \rightarrow C)$ axiome S

$\vdash ((\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A))$ axiome DN²

- Une règle d'inférence (Modus Ponens) :
$$\frac{\vdash A \rightarrow B, \vdash A}{\vdash B}$$

Exemple

Si on utilise l'ensemble d'hypothèse : $\{A \wedge B\}$ comment prouver $\vdash A \vee B$.

Forme de Hilbert :

- $A \wedge B \equiv \neg(A \rightarrow \neg B)$
- $A \vee B \equiv \neg A \rightarrow B$

Il faut trouver une dérivation :

$$\frac{\frac{\frac{}{\vdash \neg(A \rightarrow \neg B) \rightarrow (\neg B \rightarrow \neg(A \rightarrow \neg B))} K}{\vdash \neg(A \rightarrow \neg B)} \quad \vdash \neg(A \rightarrow \neg B)}{\vdash \neg B \rightarrow \neg(A \rightarrow \neg B)} \quad \text{MP} \quad \frac{\vdash \neg B \rightarrow \neg(A \rightarrow \neg B) \rightarrow (A \rightarrow \neg B) \rightarrow B}{\vdash (A \rightarrow \neg B) \rightarrow B} \quad \begin{matrix} \text{DN} \\ \text{MP} \end{matrix}$$

Commentaires sur le système de preuve

Remarques :

- Il y a d'autre variantes de ce système, utilisant d'autres règles.
- Ce minimalisme, du coup, rend ce système difficile d'emploi, il va falloir s'aider d'un logiciel pour le manipuler.
- Quelle est la relation de ce système de preuve avec la sémantique ? Il doit respecter cette sémantique il doit être **valide** i.e. tous les théorèmes générés doivent être des tautologies. A l'inverse la **complétude** doit être garantie, c'est à dire que toutes tautologies doivent être générées.

Système de déduction naturelle de Gentzen

Trois principes :

- Les preuves prennent place dans un contexte d'hypothèses
- Chaque connecteur logique à sa propre définition indépendante des autres
- Chaque connecteur est défini par des règles d'introduction (i-règles) et d'élimination (e-règles)

Règles du système de déduction de Gentzen : \wedge

- $\frac{\vdash A \quad \vdash B}{\vdash A \wedge B}$ Règle i- \wedge
- $\frac{\vdash A \wedge B}{\vdash A}$ Règle e1- \wedge
- $\frac{\vdash A \wedge B}{\vdash B}$ Règle e2- \wedge

Règles du système de déduction de Gentzen : \rightarrow

- $$\frac{\vdash A \quad \vdash A \rightarrow B}{\vdash B} \quad \text{Règle e-}\rightarrow$$

- $$\frac{\begin{array}{c} [\vdash A] \\ \dots \\ \vdash B \end{array}}{\vdash A \rightarrow B} \quad \text{Règle i-}\rightarrow$$

- Exemple : Preuve de $A \wedge B \rightarrow A$

$$\frac{\frac{[\vdash A \wedge B]}{\vdash A}}{\vdash A \wedge B \rightarrow A}$$

Un point essentiel est la portée des hypothèses qui se limite à la preuve 'en cours'.

Règles du système de déduction de Gentzen : \vee

- $\frac{\vdash A}{\vdash A \vee B}$ Règle i1- \vee

- $\frac{\vdash B}{\vdash A \vee B}$ Règle i2- \vee

- $\frac{\begin{array}{c} [\vdash A][\vdash B] \\ \vdash A \vee B \quad \vdash \overset{\dots}{C} \vdash C \end{array}}{\vdash C}$ Règle e- \vee

Ce système devient malheureusement difficile à maîtriser pour les preuves complexes.

Exemple de déduction naturelle

Calcul des séquents

- Similaire à la déduction naturelle, mais fait des hypothèses explicites
- Le séquent $\Gamma \vdash \Delta$ défini par $A_1, \dots, A_m \vdash B_1, \dots, B_n$ est vrai si $A_1 \wedge \dots \wedge A_m \rightarrow B_1 \vee \dots \vee B_n$
- Cela signifie que l'on suppose que les A_i sont vrais et l'on montre que au moins un des B_j est vrai.
- Un séquent de base est un séquent où la même formule apparaît des deux côtés du jugement, $P, B \rightarrow B, R$. Ce séquent est toujours vrai.
- Les séquents sont utilisés en 'arrière' en partant depuis la conclusion désirée jusqu'à des séquents triviaux.
- Partir en avant conduit à générer des quantités de théorèmes inutiles avec peu de chance de trouver une réponse à la preuve désirée.
- Les règles des séquents sont classées en gauches et droites, indiquant sur quel côté du jugement elles opèrent.

Calcul des séquents : Règles

$$\overline{\Gamma, A \vdash \Delta, A}^{(basic)}$$

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta} (\neg l)$$

$$\frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A} (\neg r)$$

$$\frac{\Gamma \vdash \Delta, A \quad B, \Gamma \vdash \Delta}{A \rightarrow B, \Gamma \vdash \Delta} (\rightarrow l)$$

$$\frac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \rightarrow B} (\rightarrow r)$$

$$\frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta} (\vee l)$$

$$\frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B} (\vee r)$$

$$\frac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} (\wedge l)$$

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} (\wedge r)$$

Par exemple $\rightarrow r$ correspond à l'introduction de l'implication dans la déduction naturelle

Calcul des séquents : Preuve

- Preuve de la satisfaction de la formule

$$\frac{\frac{\frac{\overline{A, B \vdash B, A} \text{ basic}}{A \vdash B, B \rightarrow A} (\rightarrow r)}{\vdash A \rightarrow B, B \rightarrow A} (\rightarrow r)}{\vdash (A \rightarrow B) \vee (B \rightarrow A)} (\vee r)$$

- Insatisfaction de la formule :

$$\frac{\frac{\overline{A \vee B \vdash B, C} \neg \text{basic} \quad \overline{B \vdash B, C} \text{ basic}}{A \vee B \vdash B, C} (\vee l)}{\frac{A \vee B \vdash B \vee C}{} (\vee r)} (\rightarrow r)$$

Calcul des séquents : Règles structurelles

$$\frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta} (\textit{weaken l})$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A} (\textit{weaken r})$$

Des règles supplémentaires permettent de réordonner, ou de dupliquer les formules, comme nous avons utilisé des ensembles ces règles non pas d'intérêt.

La règle de coupure autorise l'introduction de lemmes :

$$\frac{\Gamma \vdash \Delta, A \quad A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} (\textit{cut l})$$

Il a été prouvé que cette règle n'est pas nécessaire (très difficile à prouver) (intuitivement Γ contient autant d'information que A).

Calcul des séquents : Exercices

Prouver :

$$\neg\neg A \vdash A$$

$$A \wedge B \vdash B \wedge A$$

$$A \vee B \vdash B \vee A$$

Calcul des séquents : Exercices

Prouver :

$$((A \wedge B) \wedge C) \vdash (A \wedge (B \wedge C))$$

$$\neg(A \vee B) \vdash (\neg A \wedge \neg B)$$

Logique du 1^{er} ordre

- Extension de la logique propositionnelle afin de raisonner sur les membres de domaines non-vides
- La logique du 1^{er} ordre à des variables sur des 'individus'
- Cette logique utilise les quantificateurs \forall et \exists
- Il n'y a pas de variables sur les fonctions et prédicat ce qui existe sur les ordres supérieurs.

Termes

- Les termes représentent les 'individus' T_Σ ou Σ est la définition de la signature c'est à dire les noms des fonctions et leurs arités. Ce sont soit des fonctions n -aires et des constantes 0-aires.
- Les termes avec variables $T_\Sigma(X)$ sont construits sur l'ensemble des variables X .

Formules atomiques

- L'ensemble P des noms de prédicats avec leur arité.
- Etant donné les termes t_1, t_2, \dots, t_n et le prédicat $p \in P$, d'arité n , $p(t_1, t_2, \dots, t_n) \in FA_{\Sigma, X, P}$

Formules

- Les formules $F_{\Sigma, X, P}$ sont construites sur les formules atomiques et avec les opérateurs $\wedge, \vee, \neg, \rightarrow, \dots$ et les quantificateurs \forall et \exists , leur précedence est plus forte que les autres opérateurs.

Exemples :

$$\forall x(etudiants(x) \rightarrow jeune(x))$$

$$\forall x, y(banquier(x) \wedge vendeur(y) \rightarrow salaire(x) > salaire(y))$$

$$\forall x(etudiant(x) \rightarrow \exists y(superviseur(x, y)))$$

Triples de Pythagore :

$$\forall n, \exists i, j, k(i > n \wedge i * i + j * j = k * k)$$

Sémantique : interprétation et valuation

- Il s'agit de faire une association des termes de la logique vers des domaines d'interprétation. Une **interprétation** est donc un couple $\langle D, I \rangle$, avec l'interprétation des termes fonctionnels par des fonctions d'arité identiques $I(f) \in D^n \rightarrow D$ et des relations $I(p) \subseteq D^n$ où n est l'arité de $p \in P$.
- Il est nécessaire également de parler de **valuation** des variables X , comme une fonction de $X \rightarrow D$.
- Définition : Interprétation des termes avec variables
 - $I_V(x) = V(x)$ si x est une variable.
 - $I_V(c) = I(c)$
 - $I_V(f(t_1, \dots, t_n)) = I(f)(I_V(t_1), \dots, I_V(t_n))$

Sémantique :satisfaction

La relation $\models_{I,V}$ établit la satisfaction d'une formule.

- $\models_{I,V} p(t_1, \dots, t_n)$ si $I(p)(I_V(t_1), \dots, I_V(t_n))$ est vérifiée.
- $\models_{I,V} \neg A$ si $\models_{I,V} A$ n'est pas vérifiée.
- $\models_{I,V} A \wedge B$ si $\models_{I,V} A$ est vérifiée et $\models_{I,V} B$ est vérifiée.
- $\models_{I,V} A \vee B$ si $\models_{I,V} A$ est vérifiée ou $\models_{I,V} B$ est vérifiée.
- $\models_{I,V} \exists x A$ si il existe un $m \in D$ tel que $\models_{I,V\{m/x\}} A$ est vérifiée.
- $\models_{I,V} \forall x A$ si pour tout $m \in D$ alors $\models_{I,V\{m/x\}} A$ est vérifiée.

Remarques : Le nombre d'interprétation est infini, prouver des tautologies demande de prouver des formules pour toutes les interprétations ?

Equivalences

- $\neg(\forall x A) \equiv \exists x \neg A$ (de Morgan infinitaire)
- $\neg(\exists x A) \equiv \forall x \neg A$
- $(\forall x A) \wedge B \equiv \forall x (A \wedge B)$ x n'est pas libre dans B
- $(\forall x A) \vee B \equiv \forall x (A \vee B)$
- $(\exists x A) \wedge B \equiv \exists x (A \wedge B)$ x n'est pas libre dans B
- $(\exists x A) \vee B \equiv \exists x (A \vee B)$
- $(\forall x A) \wedge (\forall x B) \equiv \forall x (A \wedge B)$
- $(\exists x A) \vee (\exists x B) \equiv \exists x (A \vee B)$
- $(\forall x A) \rightarrow B \equiv \exists x (A \rightarrow B)$ x n'est pas libre dans B
- $(\exists x A) \rightarrow B \equiv \forall x (A \rightarrow B)$ x n'est pas libre dans B
- $(\forall x A) \equiv (\forall x A) \wedge A[t/x]$
- $(\exists x A) \equiv (\exists x A) \vee A[t/x]$

Preuve de : $\exists x (x = a \wedge P(x)) \rightarrow P(a)$

Formes Normales Prenex

Les formes normales prenex de la logique sont :

- $Q_1x_1Q_2x_2\dots Q_nx_n(A)$ où
- les Q_i sont des quantificateurs tels que \exists ou \forall
- A est une formule sans quantificateur.

Formes Normales Prenex : Exemple

Exemple : $\neg(\exists x P(x)) \wedge (\exists y Q(y) \vee \forall z P(z))$

Exemple : $(\forall x P(x)) \rightarrow (\exists y \forall z R(y, z))$

calcul des séquents pour la logique du 1^{er} ordre : \forall

$$\frac{A[t/x], \Gamma \vdash \Delta}{\forall x A, \Gamma \vdash \Delta} (\forall I)$$

$$\frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, \forall x A} (\forall r)$$

Dans $(\forall r)$ x ne doit pas être libre dans la conclusion de la règle.
Illustration par les différences entre les preuves du théorème
 $\forall x (P(x) \rightarrow P(x))$ et de la tentative de la preuve de la formule
 $(P(x) \rightarrow \forall x P(x))$

calcul des séquents pour la logique du 1^{er} ordre : \exists

$$\frac{A, \Gamma \vdash \Delta}{\exists x A, \Gamma \vdash \Delta} (\exists I)$$

$$\frac{\Gamma \vdash \Delta, A[t/x]}{\Gamma \vdash \Delta, \exists x A} (\exists r)$$

Dans $(\exists I)$ x ne doit pas être libre dans la conclusion de la règle.

calcul des séquents pour la logique du 1^{er} ordre : Exemples

- $(\forall x P(x)) \vdash (\forall z P(f(z)))$
- $(\exists x A \wedge \exists x B) \vdash (\exists x (A \wedge B))$
- $(\forall x (A \wedge B)) \vdash (\forall x A)$

Model checking : SAT solvers

Un problème SAT est un problème de décision défini par des formules logiques.

$g \in F(AP) \exists I$ une interprétation telle que $g \models_I \text{true}$

Le problème SAT se base sur une formule logique sous forme CNF.

Exemple :

- Variables $\{v_1, v_2, v_3\}$
- Formule $g = (v_1 \vee v_2) \wedge (\neg v_1 \vee v_3) \wedge (\neg v_2 \vee \neg v_1)$.
- g est satisfaisable pour $v_1 = \text{vrai}$, $v_2 = \text{faux}$, $v_3 = \text{vrai}$, alors g est logiquement vrai.

En revanche :

- $g' = (v_1 \vee v_2) \wedge (\neg v_1 \vee v_3) \wedge (\neg v_2 \vee v_1) \wedge (\neg v_2 \vee v_3) \wedge (\neg v_1 \vee \neg v_3)$
- n'est pas satisfaisable, car g' sera évalué comme faux quelles que soient les valeurs attribuées à v_1, v_2 et v_3 .

SAT solvers :Apprentissage de clauses par conflits

Déterminer si une formule sous forme normale conjonctive d'ordre n est satisfaisable est appelé problème de satisfaisabilité ou problème SAT d'ordre n , ou encore n -SAT. Selon la valeur de l'ordre n , et selon la possibilité de réduire cet ordre avec un algorithme de complexité simple, la complexité de ce problème obéira à des conditions différentes. (wikipédia)

SAT solvers :Apprentissage de clauses par conflits

Le principe des solveur de type CDCL (Conflict-Driven Clause Learning) est d'utiliser un mécanisme pour apprendre de nouvelles clauses en cours de recherche, puis d'utiliser un maximum toutes les informations apprises dans l'espoir d'améliorer le temps de recherche.

D'un point de vue pratique, cette méthodologie est très efficace pour résoudre des problèmes concrets.

SAT solvers :Méthode naïve

Soit l'instance à résoudre la formule $\phi_1 = C_1 \wedge C_2$ suivante :

$$C_1 = x_1 \vee x_2$$

$$C_2 = \neg x_1 \vee x_3$$

et l'interprétation partielle \mathcal{I} définie par :

- $x_2 = \textit{faux}$
- $x_3 = \textit{faux}$

Cette interprétation mène à un conflit car :

- x_2 implique que $x_1 = \textit{vrai}$,
- x_3 implique que $x_1 = \textit{faux}$.

SAT solvers :Méthode naïve

Pour générer une clause représentant ce conflit, il est possible de prendre la négation de l'interprétation :

$$\mathcal{I} : \neg \mathcal{I} = \neg(\neg x_2 \wedge \neg x_3) = x_2 \vee x_3.$$

Cependant, cette méthode possède un inconvénient majeur : la taille de la clause est entièrement déterminée par toute l'interprétation. Or, certaines parties de l'interprétation peuvent complètement indépendante du conflit que l'on souhaite représenter.

SAT solvers :Méthode naïve

En effet, ajoutons à l'instance ϕ_1 les formules :

$C_i = x_i \vee x_{i+1}, 4 \leq i \leq N$ et soit l'interprétation courante

$\mathcal{I} = \{x_2 = \textit{faux}, x_3 = \textit{faux}, x_{4 \leq i \leq N} = \textit{faux}\}$.

Le conflit provient des deux premières clauses et donc, des valeurs de x_2 et x_3 .

Or, la clause générée par la négation de l'interprétation prendra en compte toutes les variables à l'exception de x_1 . De ce fait, la clause sera inutilement longue.

SAT solvers :Méthodes améliorées

- Analyse des conflits
- Graphe d'implication
- Retour en arrière non chronologique
- Approches prospectives
- Recherche locale

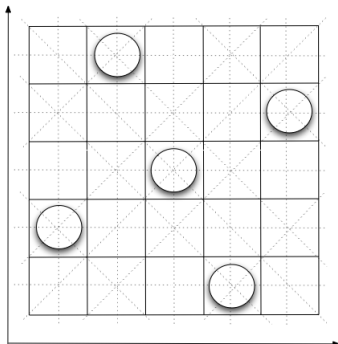
SAT solvers :Applications

Il est possible de traduire certains problèmes d'intelligence artificielle et d'utiliser les algorithmes SAT pour résoudre efficacement ces problèmes.

- Diagnostic de programmes
- Planification d'une séquence d'actions
- Model checking (séquence infinie), bounded model checking.
- Cryptographie : La complexité du problème SAT est une composante essentielle de la sécurité de tout système de cryptographie.

Résolution de problème

- Définir le problème comme un ensemble d'équation booléenne.
- Construire un système SAT.
- Le problème des 8 reines : Comment placer n reines sur un plateau de $n \times n$?



huit reines

Solving problems (cont'd)

Soit $C_{i,j}$ une constraints pour la position i,j et $Q_{i,j}$ la présence d'une reine à la position i,j :

$$C1_{i,j} = Q_{i,j} \cdot \bigwedge_{1 \leq k \leq n, k \neq i} \neg Q_{k,j} \cdot \neg Q_{k,j+i-k} \cdot \neg Q_{k,j+k-i} \cdot \bigwedge_{1 \leq l \leq n, l \neq j} \neg Q_{i,l}$$

Pas d'autres reines sur les diagonales, lignes et les colonnes.
 $C_{i,j}$ doit être satisfaite pour i and j .

$$C_{i,j} = C1_{i,j} \cdot \bigwedge_{1 \leq l \leq n} \left(\bigvee_{1 \leq k \leq n} C1_{k,l} \right)$$

Au moins une reine par colonnes.

Autres logiques : Logique intuitionniste

Logique n'utilisant pas le principe du tiers exclus. (cf wikipedia)
la loi de Peirce $((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi$ n'est pas vérifiée.

Il n'y a plus de moyen de ramener la logique à un connecteur et un quantificateur.

Chaque connecteur a sa propre sémantique.

Quelques propriétés particulières :

- $\neg A \equiv \neg\neg\neg A$ est vrai, mais
- $A \equiv \neg\neg A$ n'est pas prouvable.
- On a $A \Rightarrow \neg\neg A$ mais pas la réciproque .

Logique intuitionniste

- $f(\perp) = \perp$
- $f(A) = \neg(\neg A)$, pour une formule atomique A différente de \perp
- $f(A \wedge B) = f(A) \wedge f(B)$
- $f(A \vee B) = \neg[\neg f(A) \wedge \neg f(B)]$
- $f(A \Rightarrow B) = f(A) \Rightarrow f(B)$
- $f(\forall x P(x)) = \forall x f(P(x))$
- $f(\exists x P(x)) = \neg(\forall x \neg f(P(x)))$

Où A et B sont des formules quelconques et P est une formule ayant x comme paramètre.

Alors on a le théorème suivant :

$\Gamma \vdash_c \Delta \Leftrightarrow f(\Gamma) \vdash_i f(\Delta)$. Où \vdash_c est la déduction classique et \vdash_i est la déduction intuitionniste.

Correspondance de Curry-Howard

La correspondance de Curry-Howard, (isomorphisme de Curry-de Brouijne-Howard) :

- preuve \equiv programme
- formule \equiv type.

La correspondance de Curry-Howard était formulée par Curry pour la logique combinatoire (logique basée sur les combinatueur K et S) dès la fin des années quarante. Howard a publié en 1982 un article qui présente formellement la correspondance pour le lambda calcul simplement typé.

$$\Phi(P \Rightarrow Q) = \Phi(P) \rightarrow \Phi(Q)$$

Conclusion

- Logique : Syntaxe et Sémantique
- Preuves : différents systèmes de preuve
- Autres logiques, intuitionnisme.
- Model checking : SAT Solvers