Université d'Angers	Département Informatique
Faculté des Sciences	L3 mention Informatique, Année 04/05
Module : *Démonstration Automatique*	Jean-Michel Richer

TD n° 2
Calcul Propositionnel

On trouve en logique deux notions assez proches :

- la première appelée **démonstration** consiste à vérifier qu'une formule est un théorème. Pour ce faire on part des axiomes de la logique et on applique les règles d'inférence comme le *Modus Ponens* jusqu'à obtention du théorème ou jusqu'à ce qu'on ne puisse plus appliquer de règle d'inférence,
- la seconde qualifiée de **déduction** consiste à vérifier qu'une formule (qui n'est pas forcément un théorème) découle d'en ensemble de formules. On utilise également dans ce cas les axiomes et les règles d'inférence.

Il existe en outre un autre aspect du calcul propositionnel connu sous le nom de **satisfiabilité** d'un ensemble de clauses. On cherche dans ce cas à savoir si un ensemble de clauses du calcul propositionnel est satisfiable ou non.

Exercice 1 - On rappelle que les axiomes du calcul propositionnel sont :

(A1)	$\vdash A{\rightarrow}(B{\rightarrow}A)$	(conséquence de l'hypothèse)
(A2)	$\vdash (A{\rightarrow}(B{\rightarrow}C)){\rightarrow}((A{\rightarrow}B){\rightarrow}(A{\rightarrow}C))$	(auto distributivité de $\rightarrow$)
(A3)	$\vdash (\neg B{\rightarrow}\neg A){\rightarrow}(A{\rightarrow}B)$	(contraposition partielle)

A partir de ces axiomes montrer qu'on peut obtenir le théorème suivant :

- $\vdash A \rightarrow A$

Exercice 2 - Déduction -

A partir des axiomes du calcul propositionnel et de l'équivalence $\vdash A \rightarrow B \equiv A \vdash B$, démontrer :

- $\vdash \neg A \rightarrow (A \rightarrow B)$
- $\vdash \neg\neg A \rightarrow A$
- $A \rightarrow B, B \rightarrow C \vdash A \rightarrow C$

Exercice 3 - Déduction syntaxique et sémantique

Soit le problème suivant : *Si Jean n'a pas rencontré Pierre l'autre nuit, c'est que Pierre est le meurtrier ou Jean un menteur. Si Pierre n'est pas le meurtrier, alors Jean n'a pas rencontré Pierre l'autre nuit et le crime a eu lieu après minuit. Si le crime a eu lieu après minuit, alors Pierre est le meurtrier ou Jean n'est pas un menteur.*

- traduire le problème en calcul propositionnel
- mettre le problème sous forme clausale
- montrer que Pierre est le meurtrier en utilisant les tables de vérité
- montrer que Pierre est le meurtrier par application du principe de résolution

Exercice 4 - Opérateur de Cardinalité
L'opérateur de cardinalié, noté $\#(\alpha,\beta,p_1,p_2,\ldots,p_n)$ signifie qu'au moins α et au plus β littéraux parmi $p_1,p_2,\ldots,p_n$ doivent prendre la valeur vrai. Donnez les formes clausales correspondant aux expressions suivantes :

$\#(1,1,p)$	$\#(1,3,p,q,r)$
$\#(0,0,p)$	$\#(3,3,p,q,r)$
$\#(1,2,p,q)$	$\#(1,1,p,q,r)$
$\#(0,0,p,q)$	$\#(1,1,p_1,p_2,\ldots,p_n)$
$\#(0,1,p,q)$	$\#(0,1,p_1,p_2,\ldots,p_n)$

Dire en fonction de n combien il faut de clauses pour représenter les contraintes $\#(0,1,p_1,p_2,\ldots,p_n)$ et $\#(1,1,p_1,p_2,\ldots,p_n)$.

Exercice 5 - Satisfiabilité, *le Problème des pigeons*
On dispose d'un nombre N de pigeons et de M pigeonniers. Un pigeon doit être placé dans un pigeonnier. Un pigeonnier ne peut accueillir qu'un seul pigeon.

- traduire ce problème à l'aide de contraintes de cardinalité en utilisant une matrice p de $N \times M$ variables propositionnelles. Si la variable $p_{i,j}$ est vraie cela signifie que le pigeon i est dans le pigeonnier j.
- donnez l'expression du problème sous forme de contraintes de cardinalité pour 3 pigeons et 2 pigeonniers.
- donnez l'expression du problème précédent sous forme clausale
- résoudre ce problème en utilisant le principe de résolution
- quelle est la complexité maximale d'un algorithme permettant de résoudre le problème des pigeons (N,M) ?
- résoudre le problème des pigeons (N,M) avec le système de déduction suivant :

$$(Ext) \qquad n \geq 2 \qquad \frac{\vdash \#(\alpha_1,\beta_1,\mathcal{L}_1)\ldots\vdash \#(\alpha_n,\beta_n,\mathcal{L}_n)}{\vdash \#(\alpha_1+\ldots+\alpha_n,\beta_1+\ldots+\beta_n,\mathcal{L}_1,\ldots,\mathcal{L}_n)}$$

$$(Inc) \qquad [\alpha_1,\beta_1]\cap[\alpha_2,\beta_2]=\emptyset \qquad \frac{\vdash \#(\alpha_1,\beta_1,\mathcal{L}) \qquad \Gamma_2 \vdash \#(\alpha_2,\beta_2,\mathcal{L})}{\vdash \bot}$$