Magistère d'Informatique et Modélisation

Partiel de Logique

14 novembre 2001

Documents autorisés. Rédigez chaque partie sur une copie séparée

PARTIE 1

Exercice 1

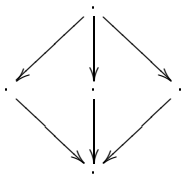
Donnez l'arbre de preuve en déduction naturelle de

$$\vdash (\varphi \vee (\varphi \Rightarrow \bot)) \Rightarrow (((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi).$$

Exercice 2

Le R-graphe d'un terme M, noté $G_R(M)$, a pour support $\{N \in \lambda \mid M \longrightarrow\!\!\!\!\!\rightarrow N\}$. Il y a un arc entre P et Q dans $G_R(M)$ si et seulement si $P \xrightarrow[R]{} Q$. S'il y a plusieurs redex dans P qui conduisent à un R-réduction menant à Q, alors il y a plusieurs flèches dans $G_R(M)$ (en toute rigueur $G_R(M)$ est un multigraphe).

1. Tracez $G_\beta(WWW)$ quand $W \equiv \lambda xy \cdot xyy$.
2. M est *fortement β-normalisable* si toute β-réduction qui démarre en M s'arrête. Montrez que si M est fortement β-normalisable, alors $G_\beta(M)$ a un support fini et M a une forme β-normale. Que pensez-vous de la réciproque à savoir que "si M est β fortement normalisable, alors $G_\beta(M)$ a un support fini et M a une forme β-normale alors M est fortement β-normalisable" ?
3. Montrez qu'il n'y a pas de terme M dont le graphe $G_\beta(M)$ est :



T.S.V.P

PARTIE 2

Exercice 3

Les énoncés suivants sont-ils des théorèmes :
– $\exists x[P(x) \to \forall y P(y)]$
– $\forall x(r(x) \to s(x)) \to (\exists x r(x) \to \forall x s(x))$.
On justifiera les résultats.

Exercice 4

Cette question se traite dans le cadre du calcul propositionnel. On appelle clause de Horn une clause C qui contient au plus un littéral positif. Elle est donc du type $\neg p_1 \vee ... \vee \neg p_n \vee q$. Si C se réduit à un littéral positif, on dit que c'est une clause positive.

L'algorithme suivant permet de déterminer si un ensemble de clauses de Horn est satisfaisable :

Soit Σ un ensemble de clauses de Horn et Σ^+ la partie de Σ constituée des clauses positives.

```
    var sat : booléen ;
sat := vrai ;
tant que (Σ+ ≠ ∅) et (sat) faire
        début soit C1 ∈ Σ+ ;
            tant qu'il existe C2 ∈ Σ − Σ+ qui coupe C1 faire
                début Σ − Σ+ := (Σ − Σ+) − {C2} ;
                      C :=résolvante(C1, C2) ;
                       si C = clausevide alors sat := faux
                       sinon si C positive alors Σ+ := Σ+ ∪ {C}
                             sinon Σ − Σ+ := (Σ − Σ+) ∪ {C}
                fin ;
            Σ+ := Σ+ − {C1}
        fin ;
si sat alors Σ est satisfaisable sinon Σ est insatisfaisable.
```

1. Tester cet algorithme sur l'ensemble

$$\Sigma = \{\neg a \vee \neg b \vee \neg d, \neg c, a \vee \neg c, b, c, d \vee \neg f, f\}.$$

2. Montrer que si Σ^+ est vide, Σ est satisfaisable. En déduire que cet algorithme termine et qu'il est de complexité polynomiale.