TD7 et révisions

Chargé de TD : Lucca Hirschi
lucca.hirschi@lsv.ens-cachan.fr
http://www.lsv.ens-cachan.fr/~hirschi

1 Fonctions (primitives) récursives

Exercice 1 (Itération)
Soit g, h des fonctions primitives récursives, montrer que la fonction $It(g,h)$ définie comme $It(g,h)(n,x) =$

```
r = g(x)
for i = 0 to n do
  r = h(r,i);
done
return r
```

est primitive récursive.

Exercice 2
Montrer que la classe des fonctions primitives récursive est la plus petite classe contenant les fonctions de base et close par composition et itération.

Exercice 3 (Fibonacci)
Montrer que la fonction définie comme suit est récursive primitive :

$$\begin{array}{rcl} f(0) & = & 1 \\ f(1) & = & 1 \\ f(n+2) & = & f(n+1) + f(n) \end{array}$$

Exercice 4
Soit f une fonction récursive à un argument. Montrer que le graphe de f est primitif récursif ssi f peut s'écire sous la forme

$$f(x) = \mu_z\{g(x,z) = 0\}$$

avec g primitive récursive.

Exercice 5
Montrer que le problème suivant est indécidable :
Donnée: f une fonction primitive récursive;
Question: f est elle identiquement nulle ?

2 Révisions

Exercice 6 (Fonctions calculables et langages récursifs)
Donner une fonction calculable dont l'image n'est pas récursive.

Exercice 7 (A court de PCP)
Montrer que PCP reste indécidable si on se restreint à des mot de longueur au plus 2. Qu'en est-il si on se retreint à des mots de longueur exactement 2 ?

Exercice 8 (Systèmes de réécriture)
Un système de réécriture de mots sur l'alphabet Σ est donné par un ensemble fini de paires $(u, u') \in \Sigma^* \times \Sigma^*$. La relation de réécriture associée est $\to_R \subseteq \Sigma^* \times \Sigma^*$ définie par

$$\omega \to_R w' \text{ ssi } \exists (u, u') \in R.\ \exists w_1, w_2 \in \Sigma^*.\ w = w_1 u w_2 \wedge v = w_1 u' w_2$$

La relation de réduction $\to_R^*$ est la clôture réflexive transitive de la relation de réécriture. Par exemple, avec

$$R = \{(c, abaca), (c, bcbabab), (aca, d), (bcb, d), (ada, d), (bdb, d)\}$$

on a $c \to_R^* d$:

$$\begin{aligned} c &\to_R abaca \to_R ababcbababa \to_R abababacababababa \to_R abababdbababa \\ &\to_R ababadababa \to_R ababdbaba \to_R abadaba \to_R abdba \to_R ada \to_R d \end{aligned}$$

Montrer que le problème d'*accessibilité* est indécidable :
Donnée: un système de réécriture R et deux mots $u, v \in \Sigma^*$;
Question: est-ce que $u \to_R^* v$?

Rappel : Si on considère deux problèmes, $\mathfrak{A}$ et $\mathfrak{B}$, on dit que $\mathfrak{A}$ se réduit à $\mathfrak{B}$ (noté $\mathfrak{A} \preccurlyeq \mathfrak{B}$) si on peut exhiber une fonction **c**alculable f qui **pour toute** instance a de $\mathfrak{A}$, renvoie une instance $b = f(a)$ de $\mathfrak{B}$, telle que *a est une instance acceptante de* $\mathfrak{A}$ **si et seulement si** *b est une instance acceptante de* $\mathfrak{B}$. **Mais**, il n'est pas nécessaire que **toutes** les instances de $\mathfrak{B}$ soient dans l'image de f !

Si on a $\mathfrak{A} \preccurlyeq \mathfrak{B}$, alors :
- si $\mathfrak{A}$ est indécidable, alors $\mathfrak{B}$ est indécidable ;
- si $\mathfrak{B}$ est décidable, alors $\mathfrak{A}$ est décidable.