Licence Informatique

Université d' Angers
2019/2020

Feuille de travaux dirigés
Systèmes de réduction

Exercice 1
Montrer que d est atteignable par réécriture à partir de $f(b, g(c))$ avec le système de réécriture :

$$\mathcal{R} = \{f(x, g(d)) \longrightarrow f(c, g(d)),\;\; g(x) \longrightarrow x,\;\; c \longrightarrow d,\;\; f(x, x) \longrightarrow x\}$$

Exercice 2 (Dérivée)
On considère des polynômes à une variable avec des coefficients entiers naturels (par exemple, $2.X^3+4.X^2+5^X+1$) et on souhaite calculer leur dérivée par rapport à X.

1. Ecrire le système de réécriture permettant de calculer la dérivée de tels polynômes.
2. Montrer les étapes de réduction pour obtenir la dérivée des polynômes $3.X^2+X+7$ et $5.X^2+3.X+4$ ainsi que de leur somme.

Exercice 3
Soit sur $\{a, b, c, d\}$ la relation $\longrightarrow$ telle que : $a \longrightarrow b$, $b \longrightarrow a$, $a \longrightarrow c$, $b \longrightarrow d$. Montrer que $\longrightarrow$ est localement confluente mais pas confluente.

Exercice 4
Ecrire un système de réécriture permettant de faire du calcul sur les formules propositionnelle. On considère la signature suivante : $\Sigma = \{et_{/2}, ou_{/2}, non_{/1}, V_{/0}, F_{/0}\}$. On souhaite donc obtenir par exemple :

$$ou(non(et(V, F)), ou(non(V), F)) \rightarrow^* V$$

Exercice 5
Avec la même signature que précédemment, écrire un système permettant de transformer une formule propositionnelle en forme normale conjonctive (c'est à dire en une conjonction de disjonctions).

Exercice 6
Etudier la confluence locale des systèmes suivants :

S_{32} :
$f(g(x), h(x)) \rightarrow a$
$g(b) \rightarrow d$
$h(c) \rightarrow d$

S_{95} :
$W(B(x)) \rightarrow I(x)$
$B(S(x)) \rightarrow S(x)$
$W(x) \rightarrow I(x)$

$$
\begin{array}{l}
S_{216} : \\
f(u(O),u(y)) \to A \\
f(v(x),v(O)) \to B \\
O \to u(O) \\
O \to v(O) \\
u(x) \to x \\
v(x) \to x \\
f(x,y) \to f(x,u(y)) \\
f(x,y) \to f(v(x),y)
\end{array}
$$

$$
\begin{array}{l}
S_{20} : \\
nats \to: (0, inc(nats)) \\
inc(: (x,y)) \to: (s(x), inc(y)) \\
hd(: (x,y)) \to x \\
tl(: (x,y)) \to y \\
inc(tl(nats)) \to tl(inc(nats))
\end{array}
$$

Comment utiliser ce dernier système S_{20} ?

Exercice 7

Ecrire un système pour manipuler les entires naturels avec 0 et la fonction successeur s d'ariité 1. Définir ensuite les opérations suivantes

— addition
— multiplication
— puissance
— factorielle
— predécesseur
— soustraction

Ecrire un prédicat *egal* qui renvoie vrai si deux expressions sont égales.

Exercice 8

Le Coffee Can Problem (Carel Scholten) est le suivant. Imaginez une tasse contenant du café en grains de deux couleurs, blancs et noirs, disposés dans un ordre quelconque. On peut ensuite disposer le contenu de la tasse en une séquence de couleurs de grains. Par exemple : white white black black white white black black.
Les règles du jeu sont données par le système suivant :

$$
\begin{array}{l}
blackwhite \to black \\
whiteblack \to black \\
blackblack \to white
\end{array}
$$

Chaque règle décrit un mouvement autorisé. Par exemple, voici une séquence de mouvements possibles (les grains soulignés participer au mouvement en cours) :

white white black <u>black white</u> white black black
white white <u>black black</u> white black black
white white white <u>white black</u> black
white white <u>white black</u> black
white <u>white black</u> black
<u>white black</u> black
<u>black black</u>
white

Le but de ce jeu est d'avoir le moins de grain possible.
Essayer de traiter une séquence de votre choix pour gagner.
Considérer à présent la règle : *whitewhite* $\to$ *white*.

Exercice 9

On veut définir des piles d'entiers. Soit les ensembles suivants :

$$
\begin{aligned}
&Zero = \{0\}\\
&Nat = Zero \cup succ(Nat)\\
&Empty = \{\Lambda\}\\
&Stack = Empty \cup push(Nat, Stack)
\end{aligned}
$$

Soit le opérateurs suivants :

$$
\begin{aligned}
&top : Stack \to Nat\\
&pop : Stack \to Stack\\
&alternate : Stack \times Stack \to Stack
\end{aligned}
$$

Donner la sémantique de ces opérations sous forme de règles de réécriture pour pouvoir prouver par exemple :

$$alternate(push(top(push(0,\Lambda)),\Lambda), pop(push(succ(0),\Lambda))) =_R push(0,\Lambda)$$