Sémantique des Langages de Programmation (SemLP) TD n° 5 : Reduction Strategies

Exercice 1 :

Check that the λ-term :

$$Pd \equiv \lambda n, f, x.n(\lambda g, h.h(gf))(\lambda y.x)(\lambda z.z)$$

represents the predecessor function where it is assumed that the predecessor of 0 is 0. Define λ-terms to represent the subtraction function, where $m - n = 0$ if $n > m$, and the exponential function n^m.

Exercice 2 : (LN : 120)

Recall the term rewriting rules of combinatory logic (CL)

$$K\ x\ y\ \to\ x \qquad\qquad S\ x\ y\ z\ \to\ x\ z\ (y\ z)$$

Using the fact that CL is a TRS prove local confluence of CL. Then adapt the method of parallel reduction presented in Section 5 of the lecture notes to prove the confluence of CL.

Exercice 3 : (LN : 137)

We recall the abstract machine for the call-by-value λ-calculus :

$$\begin{array}{rcl}
(x[\eta], s) & \to & (\eta(x), s) \\
((MM')[\eta], s) & \to & (M[\eta], r : M'[\eta] : s) \\
(v, r : c : s) & \to & (c, l : v : s) \\
(v, l : (\lambda x.M)[\eta] : s) & \to & (M[\eta[v/x]], s)
\end{array}$$

Suppose we add to the λ-calculus with call-by-value a unary operator f and a binary operator g.

1. What are the new evaluation contexts ?

2. How is the abstract machine to be modified ?

Exercice 4 : (LN : 138)

Implement the abstract machine for call-by-name (or call-by-value) using De Brujin notation.

Exercice 5 : (LN : 139)

Suppose we add to the call-by-name λ-calculus two monadic operators : C for *control* and A for *abort*. If M is a term then CM and AM are λ-terms. An evaluation context E is always defined as : $E ::= [\ ] \mid EM$, and the reduction of the control and abort operators is governed by the following rules :

$$E[CM] \ \rightarrow \ M(\lambda x.AE[x]) \ , \qquad E[AM] \ \rightarrow \ M$$

Adapting the rules of Table 15 of the lecture notes, design an abstract machine to execute the terms in this extended language (a similar exercise can be carried on for call-by-value). *Hint :* assume an operator *ret* which takes a *whole stack* and *retracts* it into a closure ; then, for instance, the rule for the control operator can be formulated as :

$$((CM)[\eta], s) \ \rightarrow \ (M[\eta], ret(s))$$