

ASCII, du caractère c . Si t_1, t_2, \dots, t_n sont des arbres, on peut représenter la liste t_1, t_2, \dots, t_n comme l'arbre $N(t_1, N(t_2, \dots, N(t_n, L\ 0) \dots))$. Enfin, on peut représenter les valeurs d'un type défini par des constructeurs dont les arguments sont eux-mêmes représentables, en numérotant les constructeurs, et en représentant la valeur $C(V_1, V_2, \dots, V_n)$ comme la liste $L\ p, t_1, t_2, \dots, t_n$ où p est le numéro du constructeur C et t_1, t_2, \dots, t_n les représentations des valeurs V_1, V_2, \dots, V_n .

On peut, en particulier, représenter ainsi le type des programmes exprimés dans l'extension de PCF et dans le langage de la machine abstraite de l'exercice 4.3. Modifier le compilateur et la machine abstraite de l'exercice 4.3 de manière à ce que les programmes écrits en PCF et en langage machine soient représentés par des arbres binaires. La machine abstraite prendra deux entrées : un programme compilé, exprimé sous la forme d'un arbre, et une valeur et appliquera ce programme à cette valeur.

Traduire le compilateur de l'exercice 4.3 en PCF. Une fois ce compilateur écrit, le compiler avec le compilateur de l'exercice 4.3. On obtient ainsi le premier compilateur exécuté par la machine abstraite de PCF. Compiler ce compilateur avec lui-même. Vérifier que le code produit est bien le même que celui obtenu avec le compilateur de l'exercice 4.3. Si c'est le cas, on peut détruire le premier compilateur et utiliser le second à la place : le compilateur a été auto-amorcé.