

TP 5 : Un interpréteur de MINIML

Le but de ce TP est d'interpréter une chaîne de caractère comme "2 + 3" en la valeur entière 5, et une autre comme "fun x -> let y = 2 in x + y" en une valeur fonctionnelle qui implémente effectivement $n \mapsto n + 2$.

Question 1. Téléchargez l'archive `miniml.tgz` et essayer de compiler et d'exécuter le programme qu'elle contient.

L'archive contient huit fichiers :

- `term.ml` : il contient une définition de l'arbre de syntaxe de MINIML et des valeurs.
- `lexer.mll`, `parser.mly`, `conv.ml` : ces fichiers servent à faire l'analyse lexicale et grammaticale permettant de transformer une chaîne de caractères en arbre de syntaxe. Ils ont été programmés par vos chargés TD ; vous ne devriez donc pas avoir à les modifier (sauf si vous voulez étendre la grammaire).
- `eval.ml` : c'est le fichier que vous devrez compléter et dans lequel vous implémenterez la fonction d'évaluation qui transformera un arbre de syntaxe en une valeur.
- `miniml.ml` : c'est le point d'entrée du programme, il contient les exemples de programmes à évaluer et lance l'évaluation.
- `Makefile` : ce dernier fichier vous permettra de compiler tous ces fichiers en tapant la commande `make`.

Question 2. En regardant les définitions des types de `term.ml` reconstituez la grammaire des termes du langage.

Question 3. L'analyse grammaticale rajoute une couche de «sucre syntaxique», ce qui explique que le programme affiché n'est pas exactement le même que le programme interprété. Quelles sont ces extensions syntaxiques ? En particulier, tâchez de bien comprendre ce qu'il se passe pour le `let rec`.

Question 4. Rajoutez dans la liste des exemples une implémentation d'une fonction de votre choix (si vous êtes à court d'idée, vous pouvez implémenter un calcul de `pgcd` ou la somme des premières puissances de deux).

Question 5. Dans le fichier `eval.ml` complétez la fonction `subst` qui effectue la substitution d'une variable par un terme clos. Assurez vous que votre fonction se comporte bien quand on substitue une variable qui est liée dans un sous-terme.

Question 6. Quel comportement suspect peut-on observer si on utilise votre implémentation pour substituer un terme contenant des variables libres ? Quelle solution apporteriez-vous (on ne vous demande pas de l'implémenter) ?

Question 7. Complétez maintenant la fonction d'évaluation. Attention : c'est ici qu'est le gros du travail. Complétez dans l'ordre donné, c'est à dire :

- les opérations sur les entiers (`TCst`, `TAdd`, `TSub`, `TMul`),
- les constructeurs qui forment la base de MINIML (`TVar`, `TApp`, `TFun`, `TLet`),
- les conditions (`TIf`, `TEq`, `TLeq`),
- les paires (`TPair`, `TLetTuple`),
- la partie impérative (`TRef`, `TDeref`, `TAss`, `TSeq`),
- la récursion avec `let rec` et `fix` (`TFix`).