

Analyse Syntaxique et Compilation, TP n° 1 : Rappel Ocaml et Introduction

Quelques pointeurs vers des docs sur ocaml

- <http://caml.inria.fr/pub/docs/manual-ocaml/index.html>, ou en Français :
- <http://www.univ-valenciennes.fr/ROI/poirriez/ensuvhc/ocamluvhc/DA-OCAML/>.

Une fonction utile :

```
val int_of_string : string -> int
```

Convert the given string to an integer.

Voici quelques fonctions du module `String`.

```
val length : string -> int
```

Return the length (number of characters) of the given string.

```
val get : string -> int -> char
```

`String.get s n` returns character number `n` in string `s`.

You can also write `s.[n]` instead of `String.get s n`.

Raise `Invalid_argument` if `n` not a valid character number in `s`.

Nous considérons dans ce TP des expressions arithmétiques préfixées de la forme $(*_{\square}4_{\square}(+_{\square}50_{\square}3))$. Ces expressions sont produites par la grammaire suivante :

$$S \rightarrow (*_{\square}S_{\square}S) \mid (+_{\square}S_{\square}S) \mid (-_{\square}S_{\square}S) \mid a$$

où a est un entier non-négatif quelconque et \square indique une espace (caractère blanc).

Exercice 1 :

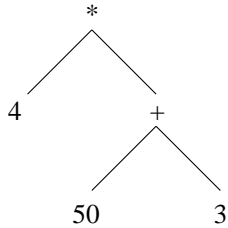
- Lesquelles des expressions suivantes sont correctes ?
 - 3
 - $(*_{\square}3_{\square}5)$
 - $(*_{\square}(+_{\square}10_{\square}1)_{\square}(-_{\square}(+_{\square}2_{\square}1)_{\square}4))$
 - $(-_{\square}(21)_{\square}(+_{\square}1_{\square}20))$
 - $(*_{\square}-10_{\square}40)$

Exercice 2 :

Écrire une fonction OCAML qui étant donné une chaîne de caractères (donc de type `string`) teste elle est bien parenthésée.

Exercice 3 :

Pour traiter une expression préfixée (par exemple pour l'évaluer, la transformer dans une autre forme, etc.) il est utile de produire une représentation intermédiaire. Nous allons utiliser un arbre. Par exemple $(*_{\square}4_{\square}(+_{\square}50_{\square}3))$ correspond à



- Définissez un type OCAML correspondant aux arbres.
- Écrire une fonction OCAML qui à partir d'une expression préfixée (chaîne de caractères) donne un arbre la représentant.
- Écrire une fonction OCAML qui évalue une expression préfixée (en passant par l'arbre).
- Écrire une fonction OCAML qui transforme une expression préfixée (donnée comme une chaîne de caractères) en expression infixée habituelle. Par exemple, l'expression infixée correspondante à $(*_{\square}4_{\square}(+_{\square}50_{\square}3))$ est $(4_{\square} *_{\square} (50_{\square} +_{\square} 3))$.
- Écrire une fonction OCAML qui transforme une expression préfixée (donnée comme une chaîne de caractères) en expression postfixée sans parenthèses. Par exemple, l'expression postfixée correspondante à $(*_{\square}4_{\square}(+_{\square}50_{\square}3))$ est $4_{\square}50_{\square}3_{\square} +_{\square} *$

Exercice 4 :

- Modifiez vos fonctions pour autoriser un nombre non fixé d'espaces
- On souhaite enlever les parenthèses des expressions préfixées Modifiez vos fonctions pour cela. Est-ce que cela est raisonnable pour une expression infixée ?