

# Travaux Pratiques de Traduction n°8

## Licence d'informatique

—2008-2009—

---

### Vérification de type

Dans ce TD nous allons voir comment définir des types et les vérifier.

---

Soit  $L$  un langage déclaratif disposant de deux types primitifs, `int` et `bool`, des expressions booléennes et entières classiques, du constructeur d'expression “?:” du C, des instructions `if` (avec le `else` optionnel), `while`, l'affectation et `print` pour afficher la valeur d'une expression. Le langage dispose également des mécanismes déclaratifs suivants :

- `var` type identificateur ;
- `var` type identificateur = expression ;
- `function` type identificateur ( liste de paramètres de la forme type nom ) = expression ;

Ecrire une grammaire LALR(1) pour  $L$ . Utiliser `bison` et `flex` pour écrire un analyseur syntaxique d'un tel langage. Vous gèrerez une table des symboles.

On s'intéresse maintenant à la vérification des types des programmes écrit en  $L$ . Ajoutez le code nécessaire à la vérification du typage dans le programme. Vous ferez attention au fait que le langage permet de définir une variable globale et un paramètre de fonction de même nom : dans le corps de la fonction, le paramètre masque la variable globale. Notez qu'on ne demande pas ici de générer du code ou de construire un AST : l'unique but du programme est de contrôler que la syntaxe et le typage est correct.