

# Travaux Pratiques de Traduction n°7 Licence d'informatique —2006-2007—

---

## Table des symboles

Dans ce TP, nous allons voir comment gérer une table des symboles. Nous nous en servons pour compiler des programmes contenant de petites fonctions arithmétiques.

---

On souhaite écrire des fonctions très simples, prenant en paramètres des valeurs entières, et retournant des expressions arithmétiques. Pour cela, on utilisera le mot-clé `function` et la syntaxe suivante :

```
function boum(x,y)=x+y;  
function rend_sept()=7;
```

On veut également pouvoir définir des variables, en utilisant le mot-clé `var` :

```
var garder=4-boum(4,2*3)*rend_sept();
```

et afficher des expressions avec le mot-clé `print` :

```
print garder+boum(7,8);
```

### ► Exercice 1. *Table des symboles*

*On veut pouvoir tester si un programme est syntaxiquement correct. Pour cela, il faut vérifier, entre autres, que les identificateurs sont correctement utilisés. Les contraintes que l'on veut respecter sont les suivantes :*

- *on ne peut pas utiliser le même identificateur pour deux fonctions/variables. Exemple de code **incorrect** :*

```
function carton(x)=45*x;  
var carton=45;
```

- on peut utiliser un même identificateur pour définir des paramètres dans des fonctions différentes. Exemple de code **correct** :

```
function trucmuche(alpha,beta)=alpha+beta;
function bidule(alpha,beta)=alpha*beta;
```

- les noms des paramètres doivent pouvoir masquer des noms de variables ou de fonctions. Exemple de code **correct** (qui devrait afficher 30) :

```
var a=14;
function boum(a,b)=a+b;
print boum(20,10);
```

- le nombre de paramètres doit être respecté. Exemple de code **incorrect** :

```
function carre(x)=x*x;
function triangle(u,v)=carre(u,v)/2;
```

- on ne doit pas utiliser les mots-clé comme noms de variables, fonctions ou paramètres. Exemple de code **incorrect** :

```
function var(print,function)=123456;
```

- On ne peut pas utiliser une variable ou une fonction non définie. Exemple de code **incorrect** (si la variable `piece_d_or` n'a pas été définie auparavant) :

```
var tresor=10000*piece_d_or;
```

1. Définissez une structure de table des symboles permettant de gérer ces différentes contraintes.
2. Écrivez les programmes `flex` et `bison` permettant de tester si un programme est syntaxiquement correct. En cas d'erreur, un message indiquant la nature de l'erreur doit être indiqué. Par exemple, l'analyse du programme suivant :

```
var tutu=60+titi;
function hypercube(y)=y*y*y*y;
function toto(x,y)=tutu()+hypercube(x,y);
```

devrait provoquer l'affichage de messages d'erreur tels que :

Ligne 1: variable 'titi' non declaree

Ligne 3: 'tutu' est une variable et non une fonction

Ligne 3: nombre de parametres incorrect pour 'hypercube'

## ► Exercice 2. Génération de code

Modifiez votre compilateur pour qu'il produise en sortie du code exécutable par la machine virtuelle.