Arnaud
FLAESCH
3I AL

Projet : Réalisation d'un pseudo-langage

Le programme a été réalisé avec win_flex et win_bison et a été testé en utilisant Cygwin64 sous Windows 8.1.

1 Fonctionnalités implémentées (avec images de leurs arbres respectifs)

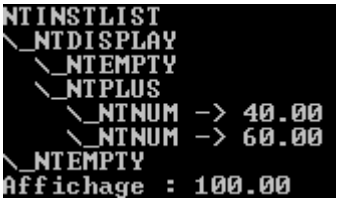
Les noms de variables doivent obligatoirement commencer par une lettre minuscule puis peuvent être suivis d'un ou plusieurs caractères alphanumériques.

Affichage d'une variable ou d'une expression

Cette fonction du langage est exécutée grâce à l'instruction « AFFICHER (Expression) ; » .

Des parenthèses doivent encadrer l'expression ou la variable.

Exemple : AFFICHER (40 + 60) ;



Conditions

Dans le cas de l'évaluation de plusieurs Expressions Booléennes, chacune d'entre elles doit être encadrée par des parenthèses et séparées par un ET ou un OU. Les mots VRAI et FAUX ont été implémenté et renvoient respectivement 1 et 0 lors de l'évaluation.

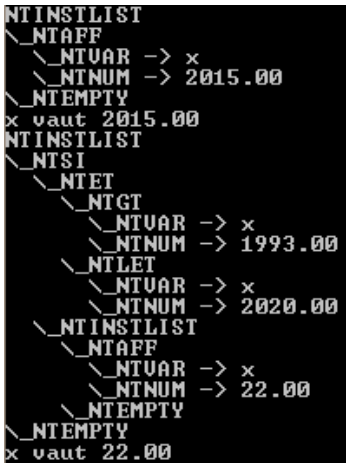
SI ALORS (Condition if then)

Cette fonction du langage est exécutée grâce à l'instruction « SI ExpressionBooléenne ; ALORS ListeInstructions ; FIN ; ».

Exemple :

x = 2015;

SI (x > 1993 ET x <= 2020) ALORS x = 22; FIN;



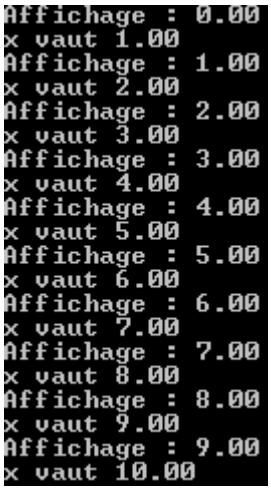
SI ALORS SINON (Condition if then else)

Cette fonction du langage est exécutée grâce à l'instruction « SI ExpressionBooléenne ALORS ListeInstructions ; SINON ListeInstructions ; FIN ; ».

Exemple : SI FAUX ALORS x = 4+6; SINON x = 10 + 10; FIN;

TANTQUE FAIRE (Boucle while)

Exemple : x = 0; TANTQUE x < 10 FAIRE AFFICHER(x); x = x+1; FIN;

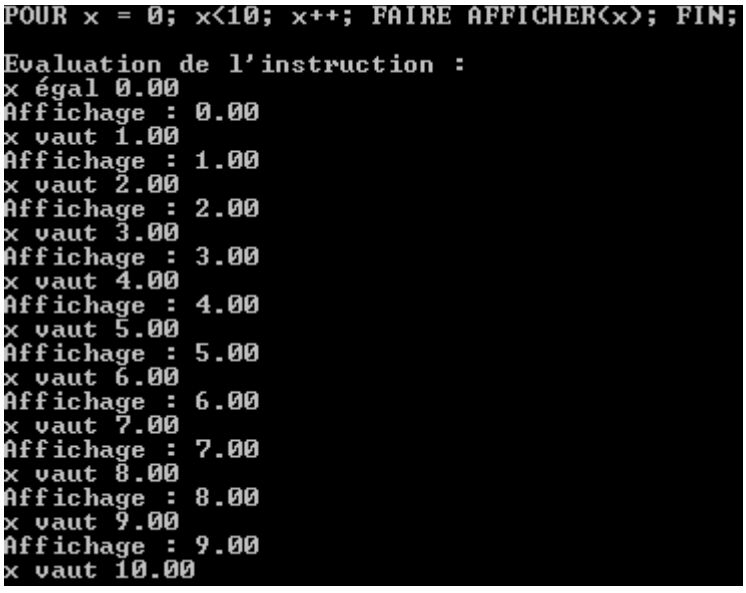


Cette boucle affiche la valeur de x puis l'incrémente, et ainsi de suite jusqu'à 10. A chaque incrémentation la nouvelle valeur de x est affichée (x vaut 1.00, etc.) car l'incrémentation correspond à un instruction d'affectation, cette dernière affichant toujours la variable et la valeur associée.

POUR Initialisation ; Condition, Incrémentation FAIRE (Boucle for)

Cette boucle reçoit en paramètre une affectation de départ, une condition et une incrémentation. Elle exécute ensuite une liste d'instructions jusqu'à ce qu'elle arrive à son terme.

Exemple: POUR x = 0 ; x<10 ; x++ ; FAIRE AFFICHER(x) ; FIN ;



Opérateurs de comparaisons

Les opérateurs plus grand que (>), plus petit que (<), inférieur ou égal (≤), supérieur ou égal (> =), égal à (==), différent de (!=), et non équivalent à (!) ont été implémentés.

Incrémentation

En tapant '++' à droite d'une variable, cette dernière voit sa valeur augmenter de 1.

Exemple : x = 0 ; x++ ;

```
x=0;

Evaluation de l'instruction :
x égal 0.00
```

```
x++;

Evaluation de l'instruction :
x vaut 1.00

Affichage de l'arbre de syntaxe :
NTINSTLIST
\_NTPLUSPLUS
  \_NTEMPTY
  \_NTVAR -> x
\_NTEMPTY
```

Affectation avec liste chaînée

Le programme utilise une liste chaînée pour stocker les couples variables/valeur et les afficher en fin d'instruction. La structure de celle-ci et ses fonctions sont déclarées dans les fichiers eval.c et eval.h.

Message d'erreur en cas de variable non initialisée

Si l'utilisateur essaye d'incrémenter une variable non définie ou même de l'afficher, un message d'erreur apparaît.

Exemple : x = x+1 ;

```
Bonjour, veuillez saisir vos instructions :
x = x+1;

Evaluation de l'instruction :
ERREUR : Variable x non initialisée !
```

2 Problèmes rencontrés

- Il est nécessaire de faire un saut de ligne après la dernière instruction du fichier en entrée sans quoi le parsing de l'instruction donne une erreur de syntaxe.
- Certains arbres ont été complexes à modéliser, notamment pour la boucle POUR et la condition SI-ALORS-SINON.