



Rapport projet LU2IN119

Interprète de langage à pile

Sira Lina ACHOURI, Marie BONBOIRE

Juillet 2023

1 Introduction

2 Projet

2.1 Définition de type

2.2 Réduction

2.3 Calculette

2.4 Dictionnaire

2.5 Evaluator

La fonction `eval : dico -> stack -> prog -> stack` est définie récursivement et considère pour les éléments du programme les différents cas suivants :

Définition

`(": ... ;")`

Afin de traiter les cas de définitions imbriquées, nous définissons une fonction auxiliaire

```
aux_def : dico -> prog -> (element list) -> (dico*(element list)*prog)
```

qui prend en paramètres un dictionnaire, un programme et une liste d'éléments contenant la nouvelle entrée du dictionnaire.

Elle retourne un triplet formé du nouveau dictionnaire, de la nouvelle entrée et du reste des éléments du programme.

Chacun des éléments du programme va être identifié comme étant soit :

- le début d'une nouvelle définition : ":"
- la fin de la définition : ";"
- le corps de la définition

Structure conditionnelle

`("... IF ... THEN" ou "... IF ... ELSE ... ENDIF")`

De même, nous traitons les cas de structures conditionnelles imbriquées, nous définissons une fonction auxiliaire

```
aux_if : prog -> stack -> (element list) -> (prog*stack)
```

qui prend en paramètres un programme, une pile et une liste d'éléments correspondant aux instructions conditionnelles.

Elle retourne un couple formé de la suite du programme et de la pile d'exécution

Chacun des éléments du programme va être identifié comme étant soit :

- la fin d'une instruction conditionnelle sans instruction alternative : "THEN"
- le fin d'une instruction conditionnelle et le début de son instruction alternative : "ELSE"
- la fin d'une instruction alternative : "ENDIF"
- le début d'une nouvelle structure conditionnelle : "IF"
- le corps d'une instruction conditionnelle

Dans chacun de ces cas, nous regardons le sommet de la pile contenant un booléen correspondant au résultat de la condition afin d'appliquer les instructions correspondantes.

Element non défini

"STRING(x)" (cf. fonction `step`)

Lorsque nous ne reconnaissons pas une structure élémentaire, nous regardons si elle n'est pas définie dans le dictionnaire.

Reste

Si nous arrivons jusqu'ici, il n'y a pas de traitement différent que celui effectué par la fonction `step` donc nous faisons appel à elle pour l'élément courant.

2.6 Programmation en PF23**2.7 Ecriture de jeux de test**