

Générateur de scripts en Machine de Turing de del Vigna

Groupe 4

Florian Jacquot
Sandra Jagodzinska
Valentina Osetrov
Agathe Wallet



Nous avons pris le script suivant pour essayer de comprendre comment procéder dans notre mission

```
% Script test basique MTdV+  
x = 0  
x = x + 1  
y = x * 2  
boucle  
  
    si (x == 2) fin }  
  
    x = 2  
  
}  
  
I  
#
```

I. Les ingrédients nécessaires

Ce qu'on va récupérer :

- Les variables présentes dans le script en MTdV+ et leurs valeurs au fur et à mesure du script → groupe 2
- Les éléments hors variables → groupe 1
- L'adresse mémoire des variables, des constantes et de la mémoire vive → groupe 3

Ce qu'on va créer :

- L'introduction d'une variable position qui définit la position actuelle de la tête sur la bande pour pouvoir se déplacer vers l'adresse des variables/constantes que l'on souhaite modifier / utiliser et vers l'adresse de la mémoire vive dans laquelle les calculs seront effectués
- Une fonction pour copier une variable/constante à une adresse donnée
- Une fonction pour comparer deux variables/constantes
- Une fonction pour générer le script en MTdV (main) 🍷🍀



II. Recette - Se déplacer sur la bande

Pour se déplacer à l'adresse mémoire d'une variable et donc la manipuler, on aura :

```
si position_actuelle < adresse_variable :  
    (adresse_variable - position_actuelle) * D  
elif position_actuelle > adresse_variable :  
    (position_actuelle - adresse_variable) * G  
else :  
    pas bouger !
```



II. Recette - Implémenter les variables

- **$x = 0$**

→ on copie 0 dans x

→ Pour les valeurs supérieures à 9, on effectuera une division par 9 avec reste et on effectuera une multiplication et addition dans la mémoire vive.

Ex : si $x = 19$, $x = 9 \times 2 + 1$

- **$x = x + 1$**

→ on copie x et 1 dans la mémoire vive puis on génère le script addition.ts avec x et 1 comme valeurs d'entrée puis on copie le résultat à l'adresse de x

→ on efface le contenu de la mémoire vive (0000000000...). Pour cela, on se place à l'adresse mémoire (en partant du principe que lecture de G à D) puis :

```
for i in range(32):  
    string += "0D"
```

- **$y = x * 2$**

→ on copie x et 2 dans la mémoire vive puis on applique le script multiplication.ts avec x et 2 comme valeurs d'entrée puis on copie le résultat à l'adresse de y

→ on efface le contenu de la mémoire vive



II. L'utilisation des ingrédients - Les conditions

- **si (x==2) fin}**
→ comparaison hors mémoire vive entre les adresses des deux variables/constantes

% on est sur x[0]
si(0) → on se déplace à 2[0]
 si(0) → on se déplace à x[1]
 si(1) → on se déplace à 2[1]
 si(1) → on se déplace à x[2]
 ...
 fin }
 } → voir cb il faut en mettre...
}



II. L'utilisation des ingrédients - Les boucles

```
boucle
  si (x==2) fin }
  x = 2
}
```



```
boucle
  % on est sur x[0]
  si(0) → on se déplace à 2[0]
  si(0) → on se déplace à x[1]
  si(1) → on se déplace à 2[1]
  si(1) → on se déplace à
x[2]
  ...
  fin }
} → voir cb il faut en
mettre...

% on retourne à x[0] (=0)
D
% permet d'arriver sur 1

% effacement de x
boucle
  si(1) 0 D}
  si(0) fin}
}

% Copie de 2 dans x
}
```

II. L'utilisation des ingrédients - L'assaisonnement

On ajoute les I P # et autres éventuels éléments, récupérés grâce à l'analyseur syntaxique et/ou au script en MTdV+



III. Le plat est prêt

On a eu la flemme de refaire un script a mano avec tout ça...

On laisse place à votre imagination...

Update éventuel à la fin du projet si notre script fonctionne...



Les contraintes, questionnements et commentaires

Pas pour l'instant mais ça ne saurait tarder...



Bon appétit !

(même si c'est nous qui allons déguster...)

