NSI T*erminale*

Langages et Programmation

TP : La machine de Turing

NOMS :
Prénoms :

Un peu d'hisoitre

Vous avez peut-être déjà entendu parler d'Alan Turing, à la tête de l'équipe qui a réussi à déchiffrer des messages de la machine Enigma pendant la deuxième guerre mondiale, donnant ainsi un net avantage aux alliés ce qui a nettement réduit la durée du conflit.
Tout juste avant l'arrivée des premiers ordinateurs, ce même Alan Turing a eu l'idée en 1936 d'une machine abstraite (donc pas une vraie qu'on peut toucher, mais juste une description théorique), assez rudimentaire, composée d'un état interne, d'un ruban sur lequel on peut écrire/lire des lettres et de règles (le programme) disant comment faire évoluer cette machine.
A priori rien de bien révolutionnaire, sauf qu'en fait si ! Les fondements que cette machine de Turing a contribué à mettre en place ont pu aboutir quelques années plus tard au concept d'ordinateurs comme vous les connaissez.

Composition de la machine de Turing

La machine de Turing est composée des éléments suivants :

- d'un ruban infini divisé en cases, dans lesquelles la machine peut écrire des symboles.
- d'une tête de lecture et d'écriture
- d'une table de transition, dont chaque ligne :
  - est associée à un état
  - spécifie les actions à effectuer quand la machine est dans cet état en fonction du symbole lu par la tête de lecture
- ayant comme actions possibles :
  - Ecrire un symbole (choisi dans un alphabet, 0 ou 1 souvent)
  - Se déplacer d'une case sur la gauche ou sur la droite
  - Changer d'état
- S'arrêtant quand on atteint un état désigné comme « final »

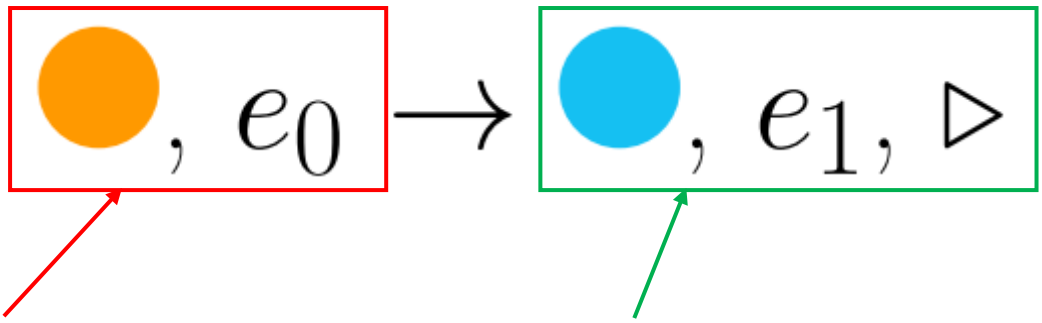
Dans cette activité :

- le ruban infini sera représenté par la plaque grise de lego
- les symboles seront représentés par les briques de lego
- un mot sera une suite de briques colorées clipsée sur le ruban les unes à côté des autres
- les différentes tables de transition (programmes) seront les documents que je vous fournirai
- une ardoise servira a écrire l'état actuel de la machine

la flèche symbolisera la tête de lecture

Les instructions

Regardons de plus près un exemple d'instruction **:**



Phase de lecture :
La brique lue est orange
alors qu'elle est dans l'état e_0

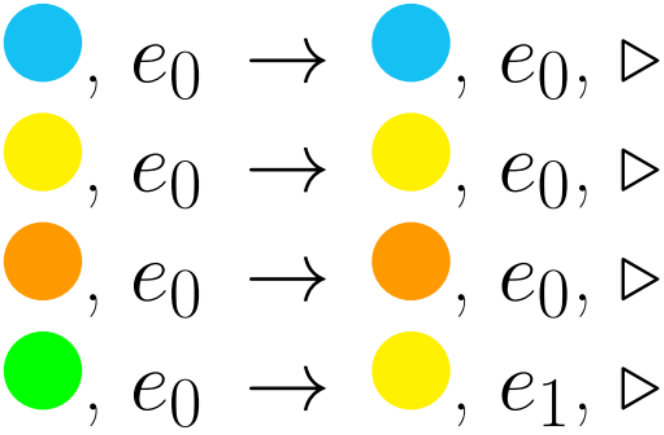
Phase d'action :
La machine remplace la brique par une bleue
la machine passe dans l'état e_1
la machine déplace la tête de lecture vers la droite

Au démarrage de la machine, celle-ci est dans l'état initial e_0 et sa tête de lecture pointe vers la brique la plus à gauche de notre ''mot''

Remarques :

- Pour représenter l'absence de brique colorée, on utilisera la couleur grise :
- Il est possible de donner une instruction valable pour toutes les couleurs de brique, on utilise alors le symbole : *
- De même, il est possible de donner une instruction valable pour tous les états possibles, on utilise alors le symbole : *
- Lorsque la machine n'a pas d'instruction correspondant à la couleur ET à l'état actuel, elle s'arrête et met fin au programme.
  On récupère alors deux choses à analyser pour comprendre ce que la machine a fait :
  le mot modifié et l'état final de la machine.

Un premier exemple



Tester la machine sur le mot : 

Puis sur le mot : 

Que fait cette machine ?

A vous de jouer

Que font les 3 machines dont je vous ai fourni les instructions ?

Machine 1 :

Machine 2 :

Machine 3 :

Défis

A vous maintenant de concevoir des machines de Turing, c'est-à-dire me donner la table d'actions pour les machines dont le fonctionnement est décrit ci-après.
Vous travaillerez sur les 3 défis de votre choix, la difficulté étant bien entendu valorisée.

1. **détecteur d'orange :**
   Entrée : un mot composé de couleurs parmi orange, jaune et bleu
   Objectif : ajoute une brique verte en fin de mot (= après la dernière brique) si le mot ne contient pas de brique orange, ne change rien sinon

2. **détecteur d'orange (2) :**
   Entrée : un mot composé de couleurs parmi orange, jaune et bleu
   Objectif : ajoute une brique verte en fin de mot (= après la dernière brique) si le mot contient une brique orange, ne change rien sinon

3. **détecteur d'orange (3) :**
   Entrée : un mot non vide composé de couleurs parmi orange, jaune et bleu
   Objectif : remplace la dernière brique par une verte s'il y a une brique orange dans le mot, ne change rien sinon

4. **détecteur d'orange (4) :**
   Entrée : un mot composé de couleurs parmi orange, jaune et bleu
   Objectif : ajoute une brique verte en début de mot (= avant la première brique) s'il y a une brique orange dans le mot, ne change rien sinon.
   Attention, dans les machines de Turing il y a toujours de l'espace libre (illimité) avant et après le mot. Il faut donc bien mettre sa tête de lecture sur la première brique du mot mais laisser de l'espace à gauche et à droite du mot sur votre ruban.

5. **parité :**
   Entrée : un mot composé de couleurs parmi orange, vert, jaune et bleu
   Objectif : l'état final doit être différent selon si le nombre de briques du mot est pair ou impair.

6. **parité verte :**
   Entrée : un mot composé de couleurs parmi orange, vert, jaune et bleu
   Objectif : idem que ci-dessus mais on doit avec l'état de contrôle à la fin savoir si le nombre de briques vertes du mot est pair ou non (les autres couleurs importent peu)

7. **pas deux de suite :**
   Entrée : un mot composé de couleurs parmi orange, jaune et bleu
   Objectif : si le mot contient deux briques de suite de la même couleur, la machine ajoute une brique verte en fin de mot

8. **pas deux de suite (2) :**
   Entrée : un mot composé de couleurs parmi orange, jaune et bleu
   Objectif : si le mot contient deux briques de suite de la même couleur, la machine remplace la deuxième brique par une verte et continue

9. **pas deux de suite (3) :**
   Entrée : un mot composé de couleurs parmi orange, vert, jaune et bleu
   Objectif : avoir en fin d'exécution un mot qui ressemble au maximum au mot d'entrée mais n'a plus deux briques de suite de la même couleur
   indice : quand on remplace une brique par une autre, attention à la nouvelle couleur choisie

10. **incrémenteur :**
    Entrée : un mot composé de couleurs parmi jaune et bleu
    Objectif : on considère qu'une brique jaune vaut 0 et une brique bleue vaut 1.
    Un mot sur le ruban est ainsi l'écriture binaire d'un nombre, par exemple 0110 est l'écriture binaire de 6 et 10111 l'écriture binaire de 23.
    Réaliser un incrémenteur (qui ajoute 1 à un nombre donné). L'algorithme est comme en décimal, sauf qu'on n'a que deux chiffres : 0 + 1 = 1 et 1 + 1 = 0 et je retiens 1. Par exemple 0100 + 1 = 0101, et 0101 + 1 = 0110 (on a propagé une fois la retenue).

Défi n° ….

Défi n° ….

Défi n° ….

Conclusion

Avec cette activité le lien avec l'informatique est clair : on exécute et on écrit des programmes, OK. Mais la question qui reste est de savoir quel est l'intérêt de continuer à utiliser ce modèle rudimentaire inventé au milieu des années 30. Pourquoi s'en souvenir plus de 80 ans plus tard alors que nous avons des ordinateurs très puissants que nous programmons dans des langages dits "évolués", bien plus proche du langage naturel ?

Parce que ce modèle, si rudimentaire soit-il, a la même capacité de calcul que n'importe quel ordinateur.

Oui nos ordinateurs modernes sont bien plus rapides, efficaces, agréables à programmer, mais ce qui est incroyable c'est que pour tout problème qu'un ordinateur sait résoudre, si complexe soit-il, on peut concevoir une machine de Turing (sans doute énorme et peu efficace) qui fait la même chose. Si vous ne voyez toujours pas l'intérêt cela va venir. Imaginez maintenant que vous trouvez un problème pour lequel vous réussissez à prouver qu'aucune machine de Turing n'est capable de le résoudre. Avec cette preuve et le fait que la machine de Turing sait faire la même chose que les ordinateurs, on obtient automatiquement le résultat suivant : aucun ordinateur, même si dans 100 ans ils sont bien plus puissants, ne pourra résoudre ce problème.

Et faire cette preuve sur les machines de Turing est rendu plus simple car le modèle de ces machines est très précis et les "instructions" ou règles sont d'un type très restreint.