

Turing's world est un petit logiciel illustrant de façon simple le fonctionnement des machines de Turing. Il permet de fabriquer, tester et faire tourner des machines de Turing des plus simples aux plus complexes. Il permet aussi d'appréhender de façon concrète les concepts de décidabilité, calculabilité, complexité, le problème de l'arrêt.

0.1 Présentation

Ergonomie

Le logiciel offre une interface graphique aux fonctionnalités usuelles (usage de la souris, menus, etc.). Il présente à l'utilisateur trois fenêtres :

1. une fenêtre qui sert à construire le diagramme des états et des transitions d'une machine de Turing. Au lancement du programme, une telle fenêtre est créée par défaut et intitulée **Untitled State Diagram**. Vous pouvez aussi ouvrir la fenêtre correspondant à l'une des machines prédéfinies (choisir **Open State Diagram** dans le menu **File**).
2. une "bande" horizontale, longue et étroite située en bas de l'écran où on écrit des symboles du langage et qui comporte à gauche un numéro indiquant l'état de la machine.
3. une petite fenêtre verticale, située à gauche, permet huit catégories d'opérations graphiques ; il suffit de sélectionner l'un des outils, puis d'aller cliquer pour définir le graphisme correspondant sur la fenêtre **State Diagram** ; de haut en bas :
 - la création d'états (appelés **n** pour node) ; par défaut le premier état créé est initial, mais cela peut être changé par la suite.
 - un transformateur qui change un état en un état acceptant (utiles pour la reconnaissance de langages et pour les machines non déterministes).
 - deux outils pour dessiner des arcs (représentant les transitions) : la seule différence est la forme des arcs.
 - un outil **T** permettant d'annoter le diagramme par un texte explicatif.
 - un outil noté par un crayon qui permet de redessiner automatiquement la fenêtre.
 - une gomme qui efface la totalité du diagramme
 - un rectangle sélecteur qui permet de sélectionner tout ou partie d'un diagramme, pour en faire une copie, couper, etc.

Utilisation de la barre des menus

La barre des menus est classique, nous ne commentons que trois cas :

- le menu **Execute** permet de faire tourner la machine (à différentes vitesses) ou même de l'arrêter si elle boucle.
- le menu **Display** permet d'afficher la machine sous forme de quadruplets, et d'afficher son temps de calcul.
- le menu **Alphabets** permet de définir l'alphabet de la bande.

T.S.V.P.

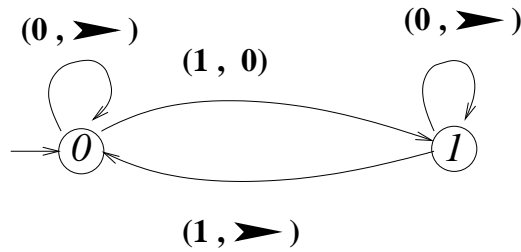


Figure 0.1 Une Machine de Turing à deux états, d'alphabet $\{0, 1, B\}$

Les transitions de la machine de la figure 0.1 sont données par les quintuplets (où \rightarrow indique un déplacement de la tête vers la droite, et *NOMOVE* pas de mouvement de la tête) :

$$(q_0, 0, 0, \rightarrow, q_0) \quad , \quad (q_1, 0, 0, \rightarrow, q_1) \quad , \quad (q_1, 1, 1, \rightarrow, q_0) \quad , \quad (q_0, 1, 0, \text{NOMOVE}, q_1)$$

L'état initial est l'état q_0 , noté 0 sur le diagramme.

On remarquera que ces transitions sont d'un type plus restreint que celles des machines vues en cours ; la machine peut : soit changer le symbole écrit sur la bande mais alors la tête ne bouge pas, soit bouger sa tête, mais alors le symbole écrit sur la bande ne change pas ; on peut donc écrire les transitions de ces machines sous forme de quadruplets, par exemple pour la figure 0.1,

$$(q_0, 0, \rightarrow, q_0) \quad , \quad (q_1, 0, \rightarrow, q_1) \quad , \quad (q_1, 1, \rightarrow, q_0) \quad , \quad (q_0, 1, 0, q_1)$$

0.2 Exercices

EXERCICE 1 Réaliser la machine de la figure 0.1 en utilisant le logiciel *Turing's world*. Que fait cette machine ? ◇

EXERCICE 2 Construire une machine de Turing qui, étant donné un nombre n représenté en notation unaire sous la forme $\underbrace{11 \dots 1}_{n \text{ fois}}^*$ fabrique la représentation binaire inversée de n : précisément, votre machine remplacera les uns par des zéros et écrira après le séparateur $*$ l'image miroir de la représentation binaire de n . Par exemple, 1111^* deviendra 0000^*001 . ◇

EXERCICE 3 Construire une machine de Turing qui, étant donné un mot $a_1 a_2 \dots a_n$ construit son image miroir : précisément, votre machine ira écrire un séparateur $\$$ à la fin du mot, puis elle remplacera les lettres du mot par des blancs et ira écrire les lettres du mot après le séparateur $\$$. Par exemple, $*001$ deviendra $*\text{BBB}\$100$, où B désigne un blanc. ◇

EXERCICE 4 Combiner les deux machines construites précédemment pour fabriquer une machine de Turing qui calcule la représentation binaire d'un nombre donné en unaire. ◇