

RAPPORT : Machine de Turing

Médéric Hurier mederic.hurier@etudiant.univ-nancy2.fr
Sabrina Arab sabrina.arab@etudiant.univ-nancy2.fr
Hamid Fnighar hamid.fnighar@etudiant.univ-nancy2.fr

Licence ISC parcours MIAGE

16 décembre 2011

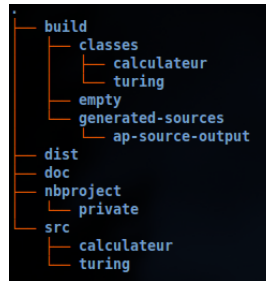


Année 2011-2012

1 Guide d'utilisation

1.1 Contenu de l'archive

L'archive contient tous les fichiers nécessaires pour comprendre et utiliser une machine de Turing.



build objets intermédiaires au format ".class"

dist exécutable de l'application au format ".jar"

doc rapport et énoncé du projet au format ".lyx" et ".pdf"

nbproject configuration de l'environnement de développement (NetBeans)

src sources du programme au format ".java"

src/calculateur paquet calculateur. Contient plusieurs machines de Turing configurées (addition binaire, division ...)

src/turing paquet turing avec les classes modélisant une machine de Turing

1.2 Exemples

Le code java peut être compilé sur votre machine, indépendamment de votre plateforme. Vous pouvez également utiliser le format distribué java. Pour cela, lancer le fichier MachineTuring/dist/MachineTuring.jar

```
mederic@gameboy:~/workspace/miage-l3/language-java/MachineTuring/dist$ java -jar MachineTuring.jar
Lanceur de machine de Turing.
Utilisation: <commande> param1 param2 param3

Commandes:
- ls: Liste toutes les machines disponibles
- use <Machine>: Sélectionne une machine pour utilisation
- run <symbole 1> <symbole 2> <symbole n>: exécute une commande sur la machine avec les symboles de l'alphabet
- help : Aide et cas d'utilisation de la machine
- config: configuration de la machine
- timing <0,1>: Enclenche la gestion du temps d'exécution (0 ou 1)
- usage : affiche le guide d'utilisation du lanceur
- exit : quitte l'application
>> █
```

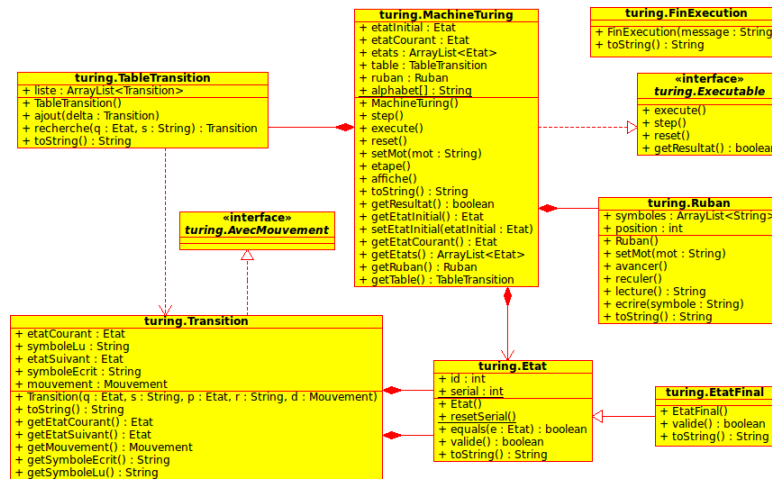
L'application fonctionne comme un interpréteur qui attend les commandes de l'utilisateur.

Suite d'instructions simples : "use Decaleur", "help", "run 0 1 0 1", "exit".

2 Conception¹

2.1 Modélisation d'une machine de Turing

Ce diagramme de classes représente une machine de Turing et ses composants



Interfaces :

- ▷ EXECUTABLE : fonctions communes que l'on attend d'une machine. Utiliser par la classe abstraite Calculateur pour implémenter le patron de conception "Adaptateur" et faciliter l'utilisation des machines.
- ▷ AVEC MOUVEMENT : Constantes de mouvement pour les transitions (GAUCHE, DROITE, IMMOBILE)

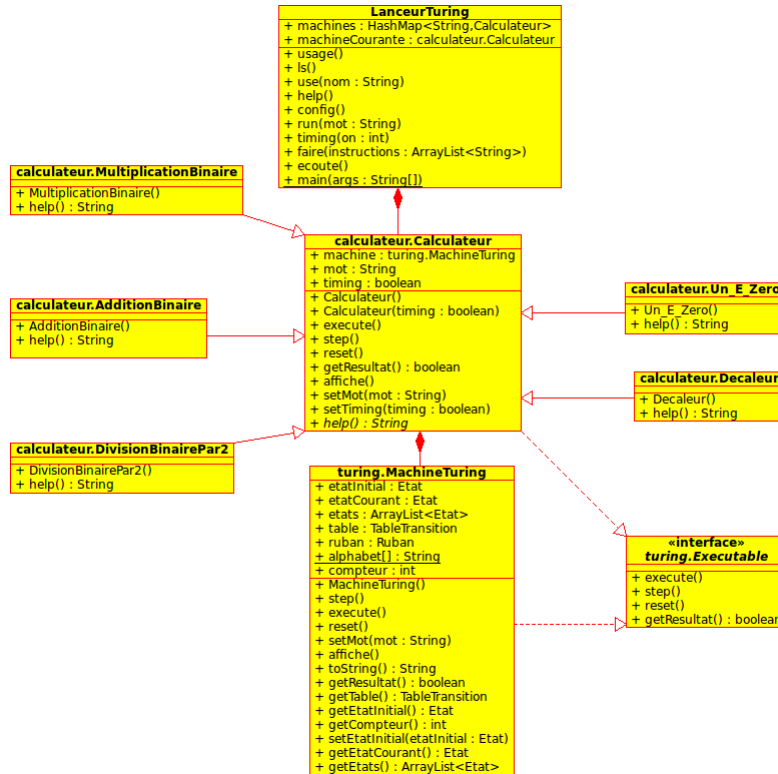
Classes :

- ▷ ETAT : État intermédiaire et non valide d'une machine. Il est auto-incrémenté à l'aide d'un compteur interne "serial".
- ▷ ETAT FINAL : classe dérivée d'un état. Les états finaux sont valides.
- ▷ TRANSITION : Fonction de transition $\delta(q,s) = (p,r,d)$
- ▷ TABLE TRANSITION : Collection de transitions pour une machine. Délègue le traitement de recherche en plus d'encapsuler la structure interne
- ▷ RUBAN : Périphérique d'entrée d'une machine composée d'une liste de symbole accessible en lecture et en écriture avec une position courante
- ▷ FIN EXECUTION : Exception levée à la fin d'exécution d'une machine (réussite ou échec)
- ▷ MACHINE TURING : Une machine est composée d'une liste d'état, d'une table de transition et d'un ruban pour un alphabet. Son état courant est stockée pour l'exécution. On peut choisir un mot, exécuter la machine et afficher son état ou sa configuration.

1. Les attributs des classes ont une visibilité public sur les diagrammes afin d'être affichés sur celui-ci

2.2 Lancement des machines

Pour faciliter l'utilisation des machines, des calculateurs sont chargés de configurer les états et les transitions



Classes :

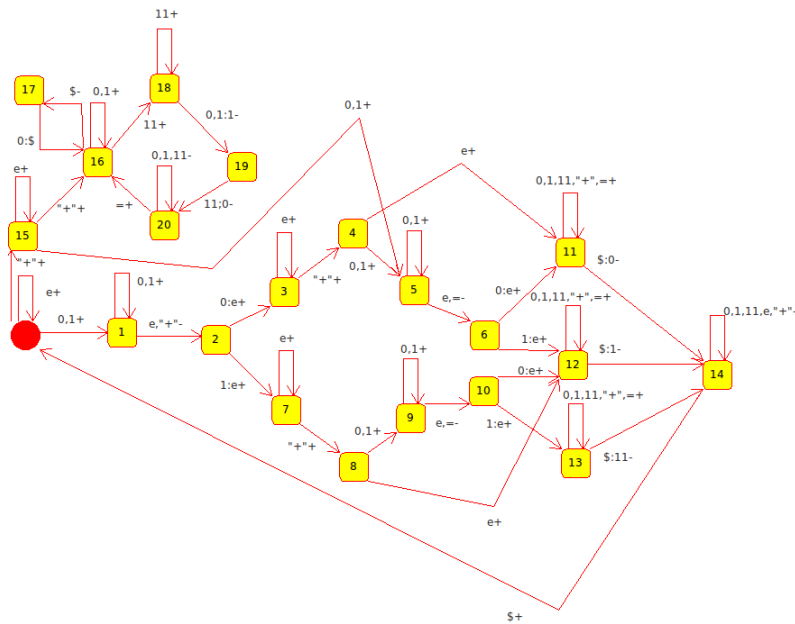
- ▷ CALCULATEUR : Classe abstraite composée d'une machine de Turing configurée. Affiche l'aide et le temps d'exécution
- ▷ DÉCALEUR : Calculateur décalant le mot d'un nombre de symbole égal à sa taille ('0 1 1' donne 'e e 0 1 1')
- ▷ UN_E_ZERO : Calculateur remplaçant les symboles d'un mot pour obtenir une suite de 1, un E et une suite de zéro ('1 0 1 E 1 1' donne '1 1 1 E 0 0')
- ▷ DIVISIONBINAIREPAR2 : Calculateur retournant la division binaire par 2 d'un mot ('1 1 0 1 0' donne '1 1 0 0')
- ▷ ADDITIONBINAIRE : Calcul l'addition binaire entre deux nombres binaires ('1 1 0 1 1 + 1 1 0 1 1 = ' donne '0 1 1 0 1 1'). Le résultat est affiché à l'envers
- ▷ MULTIPLICATIONBINAIRE : Effectue la division binaire entre deux nombres binaires ('1 0 0 x 1 1 0 1 = ' donne '1 1 0 1 0 0'). Si le premier terme est composé de plus d'un symbole '1', affiche une suite d'addition
- ▷ LANCEURTURING : Classe principale du programme. Interpréteur de commandes utilisateurs en ligne de commande.

3 Difficultés

3.1 Réalisation des machines de Turing

L'étape la plus délicate du projet était la réalisation des machines de Turing. Chaque configuration nécessite beaucoup de réflexion, et il devient vite nécessaire de faire des schémas.

Par exemple, ce diagramme d'état représente la machine permettant d'effectuer une addition binaire.



La modélisation d'une machine est pourtant simple, mais cette simplicité a une influence sur la complexité des programmes réalisés par la suite.

3.2 Mesure du temps d'exécution

Pour observer le rapidité d'une machine à traiter un mot, plusieurs méthodes sont envisageables, chacune avec ses avantages et ses inconvénients :

- Temps en millisecondes : durée réelle que met la machine à calculer le résultat. Dépend des ressources disponibles sur la machine mais permet de mesurer l'efficacité de la modélisation par analyse statistique
- Temps en nombre d'étape : permet de tester l'efficacité de la machine elle-même. Moins le nombre d'étape est important, et plus la machine est efficace

Ces deux solutions sont affichées suite à l'exécution d'un mot. Les calculs peuvent vite devenir très long selon la complexité du mot en entrée.

Par exemple, la multiplication binaire pour le mot "1 1 1 1 1 1 1 1 1 1 x 1 0 1 0 1 0 1 0 =" prend presque 40 secondes et 27682 étapes