جامعة العلوم و التكنولوجيا و الطب

Université des Sciences de Technologie et de Médecine

USTM

Rapport de Projet

Représentation Des Automates Fini En Ocaml

Réalisé Par:

- Med Said M'bareck M'bareck C09833
- Aminetou Med Abatty C09728
- Fatimetou Abdarahmane S'oeid'Ahmed C10170

Prof responsable:

Sidi Ehmety

Date de remise :25/2/2017

P l a n :

Introduction

Les mini-projets sont souvent l'occasion de voir des formations solides et plus efficaces.

Dans ce cadre ce rapport se présente pour l'explication des taches réalisées sur le mini-projet sous titre « la représentation des automates fini en ocaml ».

Plus précisément deux fonctions étaient réalisées et un programme :les fonctions sont : complet qui vérifie si un automate est complet ou non; deterministe qui vérifie si un automate est deterministe ou non et le programme dessiner_automate qui dessine un automate.

Déroulement du projet

Sender les taches puis regrouper, discuter et confirmer s'était la méthode qu'on a suivi pour résoudre le travaille demander du début jusqu'à la fin.

Une première partie est résolue très vite et facilement sans grande difficulté (les fonctions complet et detrministe) par contre une deuxième partie demandait vraiment beaucoup de réflexion , concentration et d'adjuration (le programme dessiner_automate) qui a été lui aussi en fin résolu "alhamdollah"

Deux choses importantes relatives aux codes sources:

1. La différence entre les deux types de fonctions; les fonctions (complet et deterministe) et le

Programme dessiner_automate dans le programme on a utilisé deux types l'un est utilisé par l'autre pour pouvoir associer des coordonnées a chaque état.

2. le programme dessiner_automate est conçu en 824 lignes de code avec les commentaires, ce programme n'est pas forcément très optimal, mais aussi il traite presque tous les cas (les cas des erreurs) et il n'accepte que les automates valident.

Exemples d'utilisation

Terminal

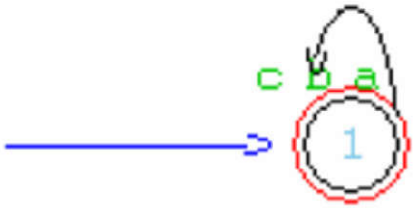
File Edit View Search Terminal Help

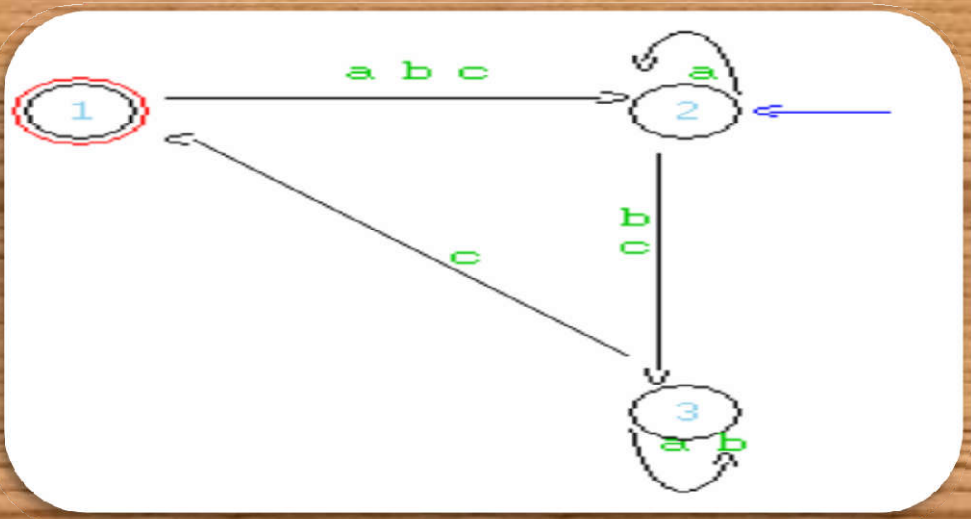
```
_--> ocamlc -o complet complet.ml
_--> ./complet
saisissez le nombre des etats de l'automate (pas les etats eux meme !!!) : 1
saisissez les etats de l'automate (entiers 1,2...) : 1
saisissez l'etat initial de l'automate (entier 1 ou 2 ou ...): 1
saisissez le nombre des etats finaux de l'automate (pas les etats eux meme !!!) : 1
saisissez les etats finaux de l'automate: 1
saisissez le nombre des lettres de l'aphabet de l'automate (pas les lettres eux meme !!!): 3
saisissez les lettres de l'aphabet de l'automate (a, b, ...): a
b
c
saisissez le nombre des transitions de l'automate (pas les transitions eux meme !!!): 3
etat de depart (ex: 1) : 1
label (ex: a) : a
etat d'arrive (ex: 2): 1
etat de depart (ex: 1) : 1
label (ex: a) : b
etat d'arrive (ex: 2): 1
etat de depart (ex: 1) : 1
label (ex: a) : c
etat d'arrive (ex: 2): 1

true
_-->
```

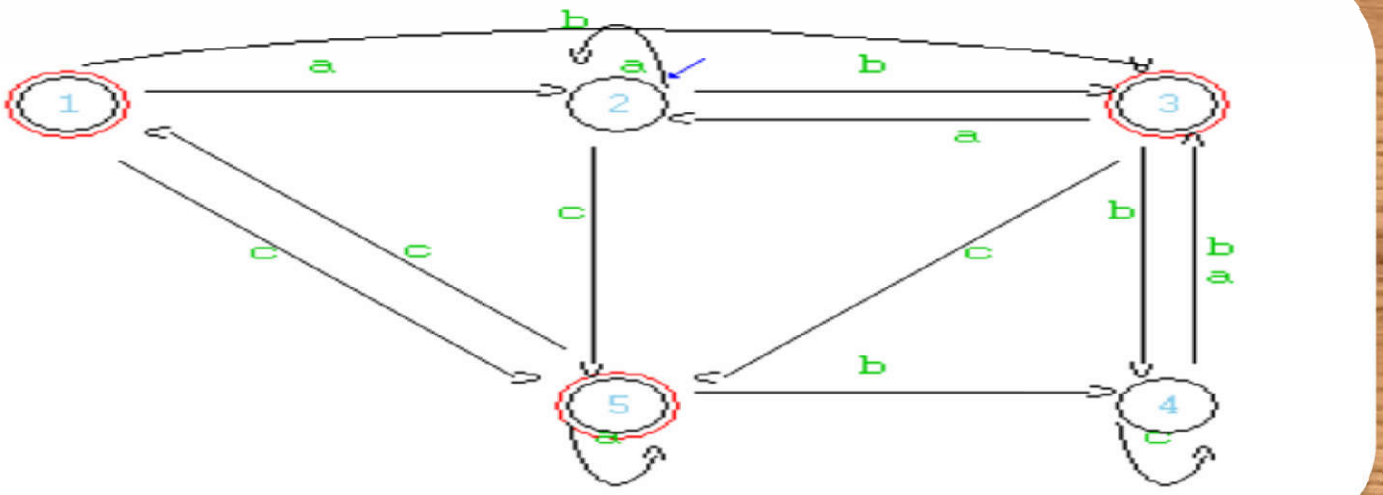
Terminal

File Edit View Search Terminal Help

```
_--> ocamlc -o deterministe deterministe.ml
_--> ./deterministe
saisissez le nombre des etats de l'automate (pas les etats eux meme !!!) : 3
saisissez les etats de l'automate (entiers 1,2...) : 1
2
3
saisissez l'etat initial de l'automate (entier 1 ou 2 ou ...): 2
saisissez le nombre des etats finaux de l'automate (pas les etats eux meme !!!) : 1
saisissez les etats finaux de l'automate: 3
saisissez le nombre des lettres de l'aphabet de l'automate (pas les lettres eux meme !!!): 1
saisissez les lettres de l'aphabet de l'automate (a, b, ...): a
saisissez le nombre des transitions de l'automate (pas les transitions eux meme !!!): 2
etat de depart (ex: 1) : 1
label (ex: a) : a
etat d'arrive (ex: 2): 2
etat de depart (ex: 1) : 1
label (ex: a) : a
etat d'arrive (ex: 2): 3

false
_-->
```







Ces testes sont effectuer sous Windows

Conclusion

Or des questions de ne pas citer notre gain Grace à ce mini-projet.

En effet on a pu approfondir nos compétences

En programmation fonctionnel (avec langage Ocaml)

Et aborder pour la première fois les graphismes dans la programmation,

Et Pour finir nous n'oublions pas à dire qu'on a vraiment sentir le gout de la programmation.

Sources

- http://www.ocaml.org/learn/
- https://caml.inria.fr/distrib/ocaml-4.03/ocaml-4.03-refman.pdf
- https://caml.inria.fr/pub/docs/manual-ocaml/libref/Graphics.html
- http://info.amatheurs.fr/index.php?page=graphisme
- https://fr.wikipedia.org/wiki/Th%C3%A9orie_des_automates
- http://www.ulb.ac.be/di/verif/tmassart/Compil/Syllabus.pdf
- https://fr.wikipedia.org/wiki/Minimisation_d'un_automate_fini_d%C3%A9terministe