Université Libre de Bruxelles

INFO-F302
Informatique Fondamentale

Rapport Informatique Fondamentale

Étudiants :
Hugo Callens
Rayan Contuliano Bravo
Ethan Rogge

Enseignants :
E. Filliot
R. Petit

25 novembre 2023

1 Modelisation d'automates en FNC

Voici quelques notations que nous utiliserons dans la suite de ce rapport :

- P représente l'ensemble des mots acceptés par l'automate
- N représente l'ensemble des mots non-acceptés par l'automate
- Σ représente l'alphabet de l'automate
- k représente le nombre au plus d'états de l'automate

Afin de modéliser un automate en FNC, il faut tout d'abord définir les variables qui seront utilisées. Pour cela, nous avons décidé de créer une variable par état de l'automate, et une variable par transition. Ainsi, pour un automate à n états et m transitions, nous aurons $n+m$ variables.

1.1 Choix des variables

1.1.1 Etats

Nous définissons notre ensemble $Q = \{q_0, q_1, \ldots, q_n | n \leq k\}$, qui représente l'ensemble des états de l'automate. Nous pouvons définir les variables e_i comme suit :

- $q_i = 1$ si l'état i est acceptant
- $q_i = 0$ si l'état i est non-acceptant

1.1.2 Etats initial

Nous définissons notre ensemble $I = \{i_0, i_1, \ldots, i_n | n \leq k\}$, qui représente l'ensemble des états initiaux de l'automate. Nous pouvons définir les variables i_i comme suit :

- $i_i = 1$ si l'état i est initial
- $i_i = 0$ si l'état i n'est pas initial

1.1.3 Transitions

Nous définissons notre ensemble $\delta = \{d_{i,j,s_i} | i, j \in Q, s_i \in \Sigma\}$, qui représente l'ensemble des transitions de l'automate. Nous pouvons définir les variables d_{i,j,s_i} comme suit :

- $d_{i,j,s_i} = 1$ si la transition avec le symbole s_i existe entre l'état i et l'état j
- $d_{i,j,s_i} = 0$ si la transition avec le symbole s_i n'existe pas entre l'état i et l'état j

1.1.4 Exemples positifs

Nous définissons $X_{p,i}$ comme étant la variable qui représente le fait que le mot p est accepté par l'automate à l'état q_i. Nous pouvons définir les variables $X_{p,i}$ comme suit :

- $X_{p,i} = 1$ si le mot p est accepté par l'automate à l'état q_i
- $X_{p,i} = 0$ si le mot p n'est pas accepté par l'automate à l'état q_i

1.1.5 Exemples négatifs

Nous définissons $Y_{n,i}$ comme étant la variable qui représente le fait que le mot n n'est pas accepté par l'automate à l'état q_i.

Nous pouvons définir les variables $Y_{n,i}$ comme suit :

- $Y_{n,i} = 1$ si le mot n n'est pas accepté par l'automate à l'état q_i
- $Y_{n,i} = 0$ si le mot n est accepté par l'automate à l'état q_i

1.2 Contraintes

— Il faut qu'il y ait un et un seul état initial

— Il faut qu'il y ait au moins un état final
— Il faut que l'ensemble P soit inclus dans l'ensemble des mots acceptés par l'automate
— Un mot se trouvant dans l'ensemble N ne peut pas être un état acceptant
— Chaque état doit avoir toutes les transitions possibles.

1.3 FNC

Voici les contraintes que nous avons définies pour modéliser un automate en FNC :

- Il faut qu'il y ait un et un seul état initial :
  - Un
    - $\bigwedge_{i \in \{1,\ldots,n\}} i_i$
  - et un seul
    - $\bigwedge_{i,j \in \{1,\ldots,n\} \wedge i \neq j} \neg i_i \vee \neg i_j$
- Il faut qu'il y ait au moins un état final :
  - $\bigvee_{i \in \{1,\ldots,n\}} q_i$
- Il faut que l'ensemble P soit inclus dans l'ensemble des mots acceptés par l'automate :
  - $\bigwedge_{p \in P} \bigwedge_{i \in Q} X_{p,i}$
- Un mot se trouvant dans l'ensemble N ne peut pas être un état acceptant :
  - $\bigwedge_{n \in N} \bigwedge_{i \in Q} (Y_{n,i})$
- Chaque état doit avoir toutes les transitions possibles (automate complet) :
  - $\bigwedge_{i \in Q} \bigwedge_{j \in Q \setminus \{i\}} \bigwedge_{s_i \in \Sigma} \bigvee_{s_j \in \Sigma} d_{i,j,s_i}$