Algorithme de McNaughton et Yamada

Elbert Nyunting   Corentin Chédotal   Ivan Dromigny

25 Janvier 2016

Table des matières

Introduction

Auteurs

Introduction
Auteurs

Robert McNaughton (1924–2014)

Mathématicien, logicien et informaticien théoricien.
Pionnier de la théorie des automates et également auteur de contributions fondamentales en plusieurs domaines d'informatique théorique, comme langages formels, les grammaires et systèmes de réécriture ainsi que la combinatoire des mots.
Il a plusieurs publications dont : *Regular expressions and state graphs for automata*, en collaboration avec Hisao Yamada.

Introduction
Auteurs

Robert McNaughton (1924–2014)

Mathématicien, logicien et informaticien théoricien.
Pionnier de la théorie des automates et également auteur de contributions fondamentales en plusieurs domaines d'informatique théorique, comme langages formels, les grammaires et systèmes de réécriture ainsi que la combinatoire des mots.
Il a plusieurs publications dont : *Regular expressions and state graphs for automata*, en collaboration avec Hisao Yamada.

Hisao Yamada (1930–2008)

Informaticien connu pour ses contributions importantes à l'informatique théorique et le développement des claviers japonais.

Introduction
L'algorithme

Introduction
L'algorithme

Brève description

Publié dans l'article *Regular expressions and state graphs for automata* en 1960 par McNaughton et Yamada.
Algorithme décrivant le passage d'un automate à état fini à l'expression rationnelle qui lui est associée.

Introduction

L'algorithme

Brève description

Publié dans l'article *Regular expressions and state graphs for automata* en 1960 par McNaughton et Yamada.
Algorithme décrivant le passage d'un automate à état fini à l'expression rationnelle qui lui est associée.

Nota

L'appellation "algorithme de McNaughton et Yamada" est parfois aussi utilisée pour parler d'un algorithme faisant le cheminement inverse (expression rationnelle vers automate ε-libre) décrit dans le même article.

Expression rationnelle

Les expressions rationnelles sur Σ décrivent les langages rationnels sur Σ. Elles sont définies ainsi :

Expression rationnelle

Les expressions rationnelles sur Σ décrivent les langages rationnels sur Σ. Elles sont définies ainsi :

Définition

- L'expression rationnelle $\emptyset$ décrit le langage rationnel $\emptyset$
- L'expression rationnelle ε décrit le langage rationnel $\{\varepsilon\}$
- $\forall a \in \Sigma$, l'expression rationnelle a décrit le langage rationnel $\{a\}$
- Soient r_1 et r_2, deux expressions rationnelles décrivant respectivement, les langages rationnels L_1 et L_2 sur Σ
  - $r_1 r_2 = r_1 | r_2 = r_1 + r_2$ décrit $L_1 L_2$
  - $r_1 \cdot r_2 = r_1 r_2$ décrit $L_1 \cdot L_2$
  - r_1^* décrit L_1^*

Transition non-passante

Transition non-passante

Définition

Une transition non passante n'admet aucune dérivation. Elle est étiquetée $\emptyset$.

Automate généralisé

Définition du cours

Automate généralisé

Définition du cours

Définition

Un automate généralisé est un automate dont les transitions sont étiquetées par des expressions rationnelles.

Automate généralisé

Détaillé

Automate généralisé

Détaillé

Plus en détails

- L'état initial possède une transition vers tous les autres états (éventuellement une *transition non passante*)
- Aucun état n'a de transition vers l'état initial
- Il existe un et un seul état final
  - distinct de l'état initial
  - qui n'a aucune transition vers les autres états
  - atteint par tous les autres états
- Tous les états (sauf initial et final) possèdent une et une seule transition vers tous les autres états

L'algorithme de MacNaughton et Yamada
Étape 0

Soit l'automate initial représenté par la la figure.

Pour rendre la figure plus lisible, nous allons représenter les arcs non passants $\emptyset$ par des traits pointillés.

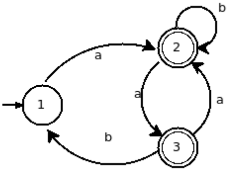

FIGURE – Automate initial

L'algorithme de MacNaughton et Yamada
Étape 1

Automate fini généralisé

- L'état initial possède une transition vers tous les autres états (éventuellement une transition $\emptyset$) ;
- Aucun état a de transition vers l'état initial

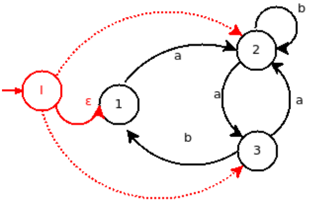

FIGURE – Étape 1

L'algorithme de MacNaughton et Yamada
Étape 1

Automate fini généralisé

- L'état initial possède une transition vers tous les autres états (éventuellement une transition $\emptyset$) ;
- Aucun état a de transition vers l'état initial
- Il n'existe un et un seul état final distinct de l'état initial, sortant vers aucun autre état et atteignable par tous les autres états

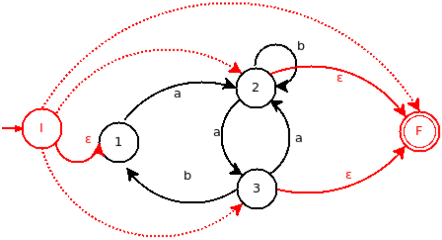

FIGURE – Étape 1

L'algorithme de MacNaughton et Yamada
Étape 1

Automate fini généralisé

- L'état initial possède une transition vers tous les autres états (éventuellement une transition $\emptyset$) ;
- Aucun état a de transition vers l'état initial
- Il n'existe un et un seul état final distinct de l'état initial, sortant vers aucun autre état, atteignable par tous les autres états
- Tous les états (sauf initial et final) possèdent une **et une seule** transition vers tous les états.

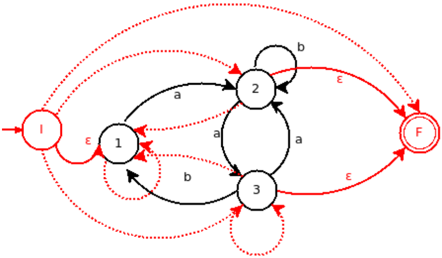

FIGURE – Étape 1

L'algorithme de MacNaughton et Yamada
Étape 2

Élimination d'un état ni initial ni final q_e.

La réduction consiste à éliminer **successivement** les états non initial et final.

- Nous allons enlever l'état 1 tout en gardant le langage.

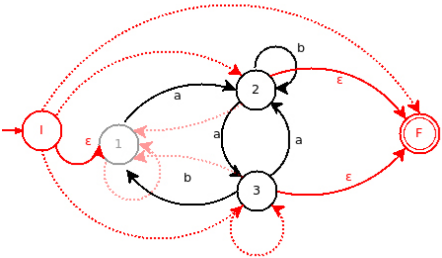

FIGURE – Étape 2

L'algorithme de MacNaughton et Yamada
Étape 2

Élimination d'un état ni initial ni final q_e.

La réduction consiste à éliminer **successivement** les états non initial et final.

- Nous allons enlever l'état 1 tout en gardant le langage.
- Ici, *I 1 2* est remplacé par la transition *a* et donc *I 2 3* est maintenant *ab* a*.

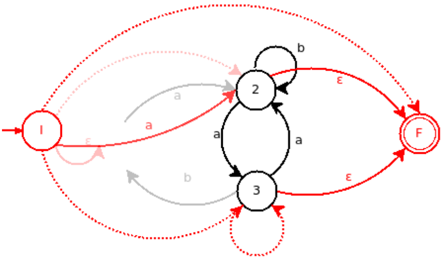

FIGURE – Étape 2

L'algorithme de MacNaughton et Yamada
Étape 2

Élimination d'un état ni initial ni final q_e.

La réduction consiste à éliminer **successivement** les états non initial et final.

- Nous allons enlever l'état 1 tout en gardant le langage.
- Ici, *I 1 2* est remplacé par la transition *a* et donc *I 2 3* est maintenant *ab* a*.
- Ensuite *3 2 3* aura comme transition *(ba|a)b* a*

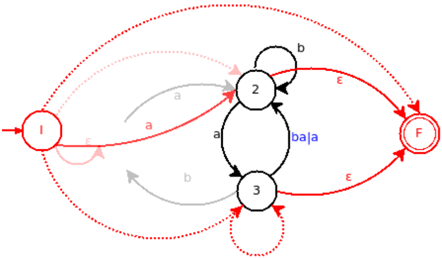

FIGURE – Étape 2

L'Algorithme de MacNaughton et Yamada

Etape 3

Élimination de l'état 2.

- Maintenant débarrassons nous de l'état 2.

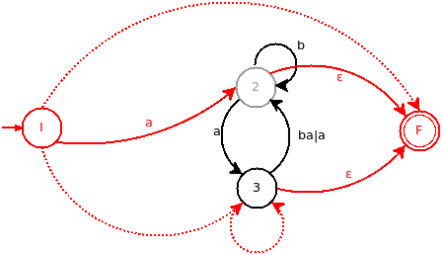

FIGURE – Étape 3

L'Algorithme de MacNaughton et Yamada

Etape 3

Élimination de l'état 2.

- Maintenant débarrassons nous de l'état 2.
- Nous avons donc *I F* qui remplace *I 2 F* avec comme transition *ab** et qui "concrétise" la transition ∅ partant de I vers F

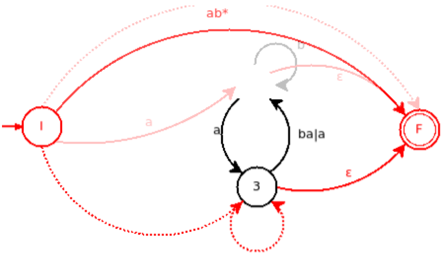

FIGURE – Étape 3

L'Algorithme de MacNaughton et Yamada
Etape 3

Élimination de l'état 2.

- Maintenant débarrassons nous de l'état 2.
- Nous avons donc *I F* qui remplace *I 2 F* avec comme transition *ab** et qui "concrétise" la transition ∅ partant de I vers F
- La voie *I 2 3*, qui était *ab* a* passe directement de *I* à *3* par la transition *ab* a*

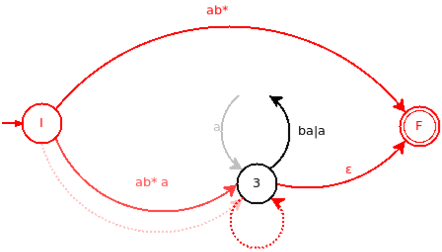

FIGURE – Étape 3

L'Algorithme de MacNaughton et Yamada

Etape 3

Élimination de l'état 2.

- Maintenant débarrassons nous de l'état 2
- Nous avons donc *I F* qui remplace *I 2 F* avec comme transition *ab** et qui "concrétise" la transition ∅ partant de I vers F
- La voie *I 2 3*, qui était *ab* a* passe directement de *I* à *3* par la transition *ab* a*
- De même, la boucle sur 3 est maintenant *(ba|a)b* a*

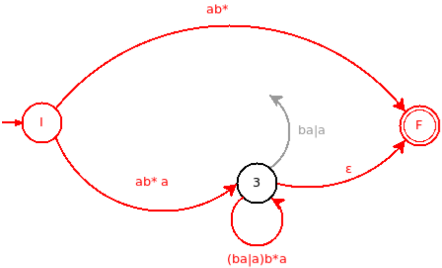

FIGURE – Étape 3

L'Algorithme de MacNaughton et Yamada
Etape 3

Élimination de l'état 2.

- Maintenant débarrassons nous de l'état 2
- Nous avons donc *I F* qui remplace *I 2 F* avec *ab**, la transition $\emptyset$ partant de I vers F
- La voie *I 2 3*, qui était *ab* a* passe directement de *I* à *3* par la transition *ab* a*
- De même, la boucle sur 3 est maintenant $(ba|a)b^*\ a$
- Et enfin, *I 3 F* est maintenant $(ba|a)b^*|\epsilon$

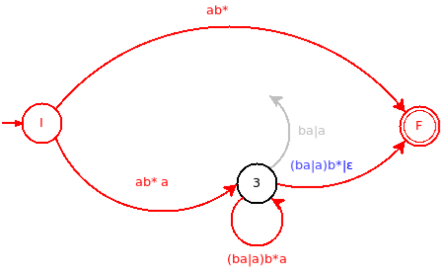

FIGURE – Étape 3

L'Algorithme de MacNaughton et Yamada
Fin

Élimination de l'état 3.

- Pour terminer, nous allons éliminer l'état 3

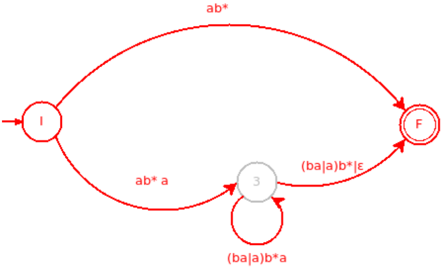

FIGURE – Fin

L'Algorithme de MacNaughton et Yamada

Fin

Élimination de l'état 3.

- Pour terminer, nous allons éliminer l'état 3
- *I F* aura donc comme transition *ab*a((ba|a)b*a)* ((ba|a)b*|ϵ)*

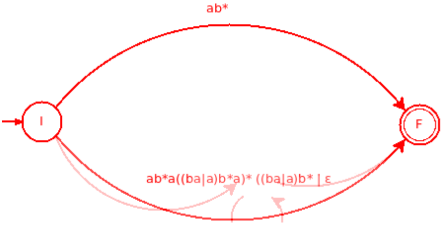

FIGURE – Fin

L'Algorithme de MacNaughton et Yamada

Fin

Élimination de l'état 3.

- Pour terminer, nous allons éliminer l'état 3
- $I\ F$ aura donc comme transition $ab^*a((ba|a)b^*a)^*\ ((ba|a)b^*|\epsilon)$
- L'expression résultant est donc $\left(\mathrm{ab^*a((ba|a)b^*a)^*\ ((ba|a)b^*|}\epsilon)\right)\ |\ \mathrm{ab^*}$

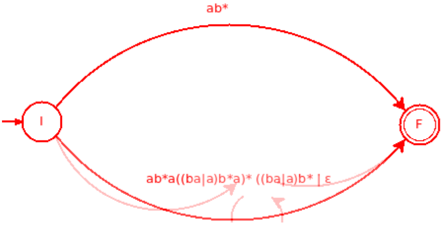

FIGURE – Fin

Références bibliographiques

Références bibliographiques

Livres

- WOLPER, Pierre. *Introduction à la calculabilité.* 2006 – Disponible à la BU

Références bibliographiques

Livres

- WOLPER, Pierre. *Introduction à la calculabilité.* 2006 – Disponible à la BU

Cours

- AMSILI, Pascal. *Langages rationnels.* Université Paris Diderot : Master de Linguistique Informatique. Février 2014 – Disponible en ligne, consulté le 23 Janvier 2016
- ENGUEHARD, Chantal et MONFROY Eric. *X6I0020 – Fondements de calculs et de calculabilité.* Université de Nantes : Licence 3 Informatique. 2016/2017 – Disponible sur Madoc, consulté le 23 Janvier 2016