Logique avancée

4 - Automates alternants

Master Info Nice Sophia Antipolis
E. Lozes

29. avril 2019

Contenu de la séance

- Automates alternants
- Définition
- Non-déterminisation
- Complémentation

Ange et démon

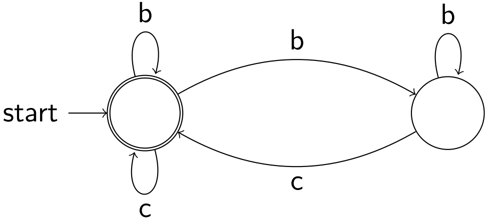
Qu'est-ce qu'une machine non-déterministe ?

- une machine qui admet plusieurs exécutions pour un même mot ; au cours de la lecture du mot, plusieurs choix sont possibles pour avancer
- les décisions peuvent être prises par un « allié » qui aide à accepter le mot
- on parle parfois de non-déterminisme « angélique »
- les NFA sont des machines non-déterministes « angéliques »

mais il existe aussi un non-déterminisme « démoniaque »

Non-déterminisme démoniaque

Notons $L^{-}(\mathcal{A})$ l'ensemble des mots w tels que toutes les exécutions de $\mathcal{A}$ sur w terminent par un état acceptant.



- $L^{-}(A) = \epsilon + (b+c)^{*}.c$
- en non-déterminisme "angélique" $L(A) = (b+c)^{*}$

Dualité du non-déterminisme

Soit $\mathcal{A} = (Q, \Sigma, \delta, q_I, F)$. On définit l'automate $\overline{\mathcal{A}}$ dual de $\mathcal{A}$ comme suit.

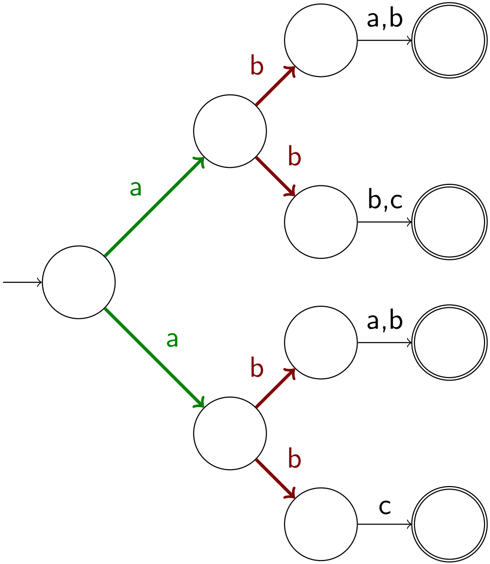
$$\overline{\mathcal{A}} \triangleq (Q, \Sigma, \delta, q_I, Q \backslash F)$$

<u>Proposition</u> : $w \in L(\mathcal{A}) \quad \text{ssi} \quad w \notin L^{-}(\overline{\mathcal{A}})$

Automates alternants

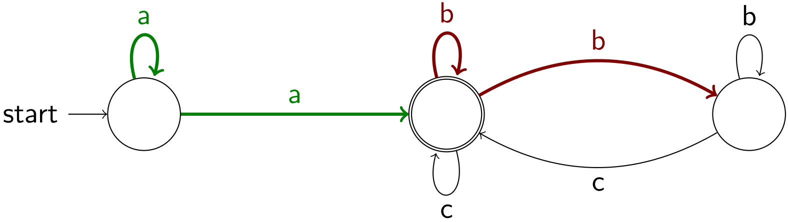
le prouveur prend les décisions pour les choix angéliques

le contradicteur prend des décisions pour les choix démoniaques

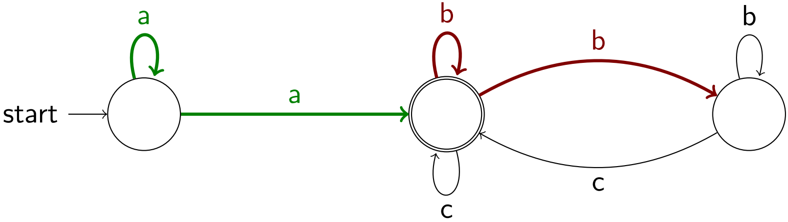


$$
\begin{aligned}
& L(A) \\
= \;& a.b\big((a+b)\cap(b+c)\big) \\
& \;+ a.b\big((a+b)\cap c\big) \\
= \;& abb + \emptyset \\
= \;& abb
\end{aligned}
$$

Exercice : L(A)= ?



Exercice : L(A)= ?



$$L(A) = a^{+} \ + \ a^{+}.(b+c)^{*}.c$$

Automates alternants : définition formelle

Formule booléene positive

Étant donné un ensemble Q, l'ensemble $\mathbb{B}^+(Q)$ des formules booléenes positives est l'ensemble des formules *sans négation* de la logique propositionnelle ayant pour ensemble de variables propositionnelles Q.

Autrement dit

- $Q \subseteq \mathbb{B}^+(Q)$: tout élement q de Q est une formule
- si f, g sont des formules, alors $f \vee g$ et $f \wedge g$ sont aussi des formules

exemple : $Q = \{q_1, q_2, q_3\}$, $f = (q_1 \wedge q_2) \vee q_3$

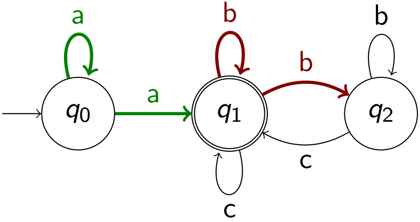
Automates alternants : définition formelle (2)

Définition

Un automate fini alternant (AFA) est un tuple $\mathcal{A} = (Q, \Sigma, \delta, q_I, F)$, où

$$\delta : Q \times \Sigma \to \mathbb{B}^+(Q).$$

exemple :



δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

Exécution et condition d'acceptance d'un AFA

Si $M \subseteq Q$ et $f \in \mathbb{B}^+(Q)$, on note $M \models f$ la relation de satisfaction

$$
\begin{array}{lcl}
M \models q & \iff & q \in M \\
M \models f \vee g & \iff & M \models f \text{ oder } M \models g \\
M \models f \wedge g & \iff & M \models f \text{ und } M \models g
\end{array}
$$

<u>exemple</u> : pour $f = q_1 \wedge (q_2 \vee q_3)$, on a

$$
\begin{array}{r}
\{q_1, q_2\} \models f \\
\{q_1, q_3\} \models f \\
\{q_2, q_3\} \not\models f \\
\{q_1\} \not\models f
\end{array}
$$

Exécution d'un AFA

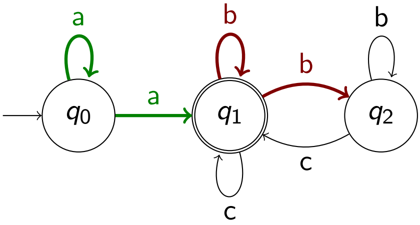
L'exécution d'un AFA $\mathcal{A} = (Q, \Sigma, \delta, q_I, F)$ sur le mot $w = a_1 \dots a_n \in \Sigma^*$ est un arbre

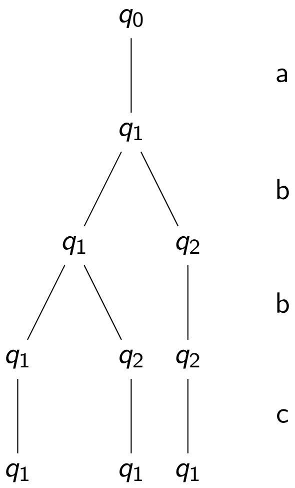
1. fini, enraciné, non-ordonné (i.e. un graphe acyclique connexe avec un sommet de degré 1 pris comme racine)
2. Q-étiqueté (chaque sommet v a une étiquette $\lambda(v) \in Q$).
3. toutes les branches sont de longueur n
4. l'étiquette de la racine est l'état initial q_I
5. pour tout sommet de profondeur i et d'étiquette q, si ses fils ont pour étiquettes $q_1, \dots, q_k$, alors $\{q_1, \dots, q_k\} \models \delta(q, a_{i+1})$

L'exécution est acceptante si toutes les feuilles de l'arbre sont étiquetées par des états acceptants.

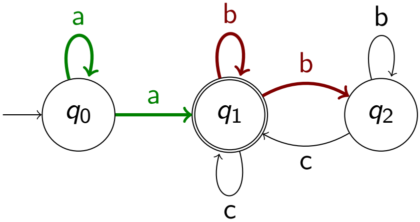
Le langage $L(\mathcal{A})$ reconnu par $\mathcal{A}$ est l'ensemble des mots pour lesquels il existe une exécution acceptante.

Exemple d'exécution acceptante : $w = abbc$

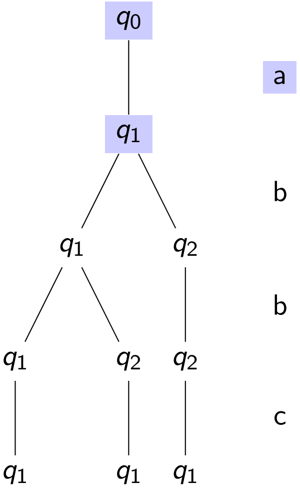




Exemple d'exécution acceptante : $w = abbc$

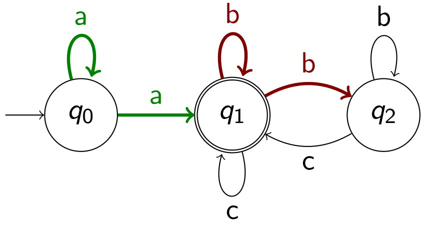


δ	a	b	c
q_0	$\boxed{q_0 \vee q_1}$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

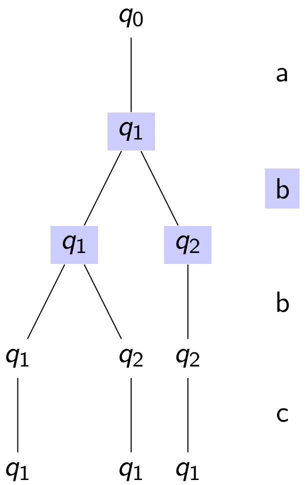


$$\{q_1\} \models q_0 \vee q_1$$

Exemple d'exécution acceptante : $w = abbc$

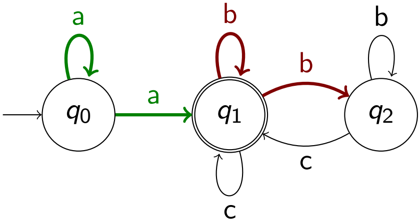


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

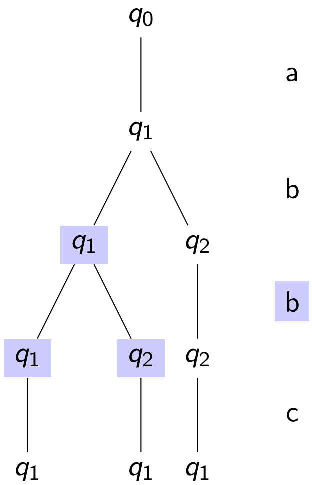


$$\{q_1, q_2\} \models q_1 \wedge q_2$$

Exemple d'exécution acceptante : $w = abbc$

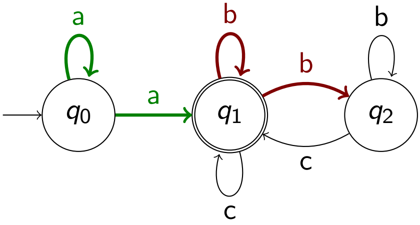


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

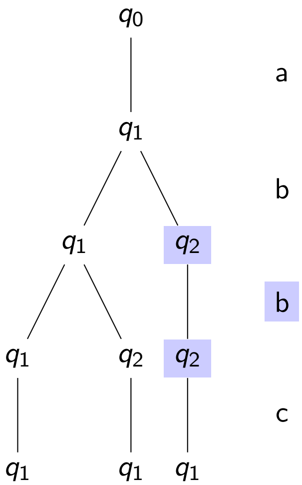


$$\{q_1, q_2\} \models q_1 \wedge q_2$$

Exemple d'exécution acceptante : $w = abbc$

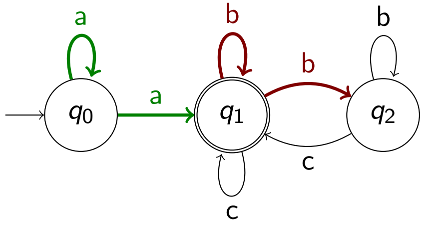


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

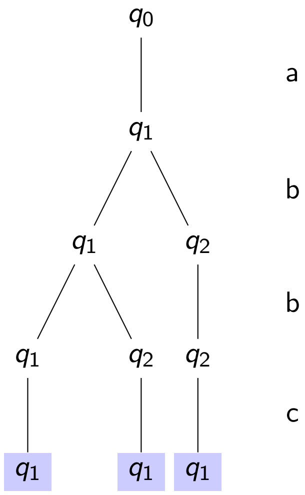


$$\{q_2\} \models q_2$$

Exemple d'exécution acceptante : $w = abbc$



δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$



$$\{q_1, q_1, q_1\} \subseteq F$$

Faisons le point

- si L est regulier, il est reconnaissable par un AFA (puisque un NFA est un cas particulier d'AFA)
- si L_1, L_2 sont reconnaissables par des AFAs, alors $L_1 \cup L_2$ et $L_1 \cap L_2$ aussi

questions :

- peut-on seulement reconnaitre des langages réguliers avec des AFA ?
- les langages reconnaissables par AFA sont-ils clos par complément ?

Dualité

Formule duale

$$\begin{array}{rcl} \overline{q} & \triangleq & q \\ \overline{\varphi_1 \vee \varphi_2} & \triangleq & \overline{\varphi_1} \wedge \overline{\varphi_2} \\ \overline{\varphi_1 \wedge \varphi_2} & \triangleq & \overline{\varphi_1} \vee \overline{\varphi_2} \end{array}$$

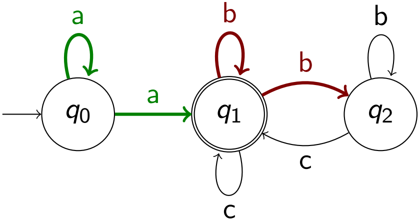
Automate dual

Soit $\mathcal{A} = (Q, \Sigma, \delta, q_I, F)$ un AFA. L'AFA $\overline{\mathcal{A}}$ dual de $\mathcal{A}$ est défini comme suit.

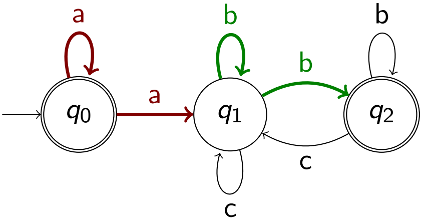
$$\overline{\mathcal{A}} \triangleq (Q, \Sigma, \overline{\delta}, q_I, Q \backslash F)$$

où $\overline{\delta}(q, a) \triangleq \overline{\delta(q, a)}$.

Exemple



$\Updownarrow$ dual



Attention ! L'état puits (non représenté) devient un état acceptant

Complémentation d'un AFA

Théorème : soit $\mathcal{A}$ un AFA et $\overline{\mathcal{A}}$ son dual. On a

$$L(\overline{\mathcal{A}}) = \Sigma^* \backslash L(\mathcal{A}).$$

Démonstration (1)

Notations utiles

Fonction de transition généralisée

On définit la fonction

$$\hat{\delta} : \mathbb{B}^+(Q) \times \Sigma^* \to \mathbb{B}^+(Q)$$

par récurrence sur la longueur du mot et sur la formule booléenne positive comme suit

$$\begin{array}{lcl}
\hat{\delta}(q, \varepsilon) & \triangleq & q \\
\hat{\delta}(q, aw) & \triangleq & \hat{\delta}(\delta(q, a), w) \\
\hat{\delta}(\varphi_1 \vee \varphi_2, w) & \triangleq & \hat{\delta}(\varphi_1, w) \vee \hat{\delta}(\varphi_2, w) \\
\hat{\delta}(\varphi_1 \wedge \varphi_2, w) & \triangleq & \hat{\delta}(\varphi_1, w) \wedge \hat{\delta}(\varphi_2, w)
\end{array}$$

Démonstration (2)

Lemmes utiles

<u>Lemme 1</u> : $w \in L(A) \quad \text{ssi} \quad F \models \hat{\delta}(q_I, w)$.

<u>Lemme 2</u> : $M \models \varphi \quad \text{ssi} \quad Q \backslash M \not\models \overline{\varphi}$.

le théorème découle de ces deux lemmes :

$$
\begin{array}{ll}
 & w \notin L(A) \\
\text{ssi} & F \not\models \hat{\delta}(q_I, w) \\
\text{ssi} & Q \backslash F \models \overline{\hat{\delta}(q_I, w)} \\
\text{ssi} & Q \backslash F \models \hat{\overline{\hat{\delta}}}(q_I, w) \\
\text{ssi} & w \in L(\overline{\mathcal{A}})
\end{array}
$$

Lien avec les langages réguliers

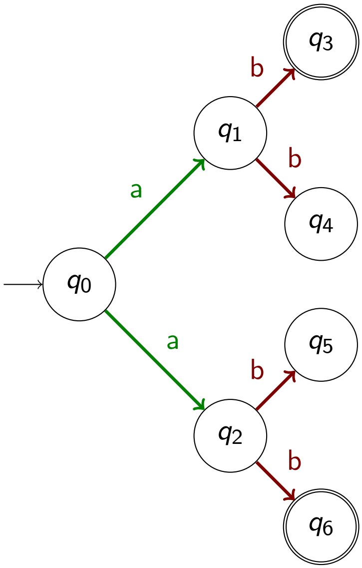
Théorème : Pour tout AFA $\mathcal{A}$ il existe un DFA $\mathcal{A}'$ avec $2^{2^{|\mathcal{A}|}}$ états tel que $L(\mathcal{A}) = L(\mathcal{A}')$

preuve :
Notons modeles$(\varphi) = \{M : M \models \varphi\}$ et $\varphi \sim \psi$ l'équivalence entre formules définie par modeles$(\varphi) =$ modeles(ψ)

Alors $\mathbb{B}^+(Q)/\sim$ contient $2^{2^{|\mathcal{A}|}}$ classes d'équivalences, chacune caractérisée par son ensemble de modèles.

On pose $\mathcal{A}' = (\mathbb{B}^+(Q)/\sim, \Sigma, \hat{\delta}, q_I, \{[\varphi]_\sim : F \models \varphi\})$.
C'est un DFA équivalent à $\mathcal{A}$!

Exemple



Exemple

$$\begin{tikzcd} {} \arrow[r] & \boxed{q_0} \arrow[r, "a"] & \boxed{q_1 \vee q_2} \arrow[r, "b"] & \boxed{(q_3 \wedge q_4) \vee (q_5 \wedge q_6)} \end{tikzcd}$$

Non-déterminisation

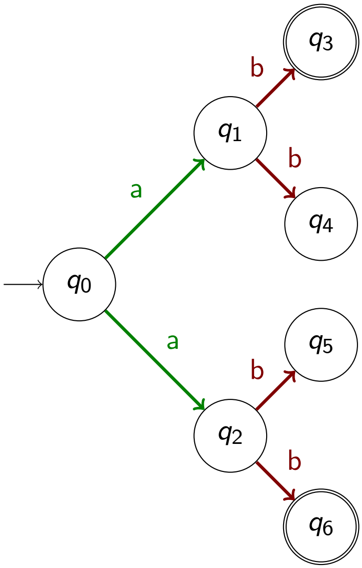
<u>Théorème</u> : pour tout AFA $\mathcal{A}$ il existe un NFA $\mathcal{A}'$ avec $2^{|\mathcal{A}|}$ états tel que $L(\mathcal{A}) = L(\mathcal{A}')$

<u>preuve</u> : soit $\mathcal{A}' = (2^Q, \Sigma, \{q_I\}, \Delta, F')$ défini comme suit :

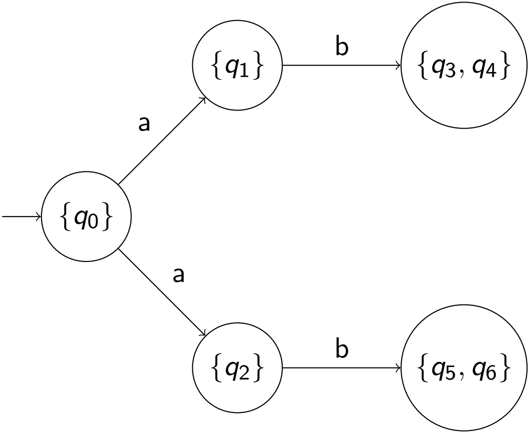
- $F' = \{M : F \models \bigwedge_{q \in M} q\} = \{M : M \subseteq F\}$
- $(M, a, M') \in \Delta$ ssi $M' \models \bigwedge_{q \in M} \delta(q, a)$

Cet automate reconnait bien le même langage que $\mathcal{A}$ par propriété d'amnésie (présentée dans 3 transparents)

Exemple



Exemple



Discussion

On pourrait envisager de traduire une formule de MSO en un AFA : les formules de base $x < y$ et $X(y)$ donnent les automates vus précédemment, et les opérations booléenes sur les formules sont reflétées par les opérations booléenes sur les AFA.

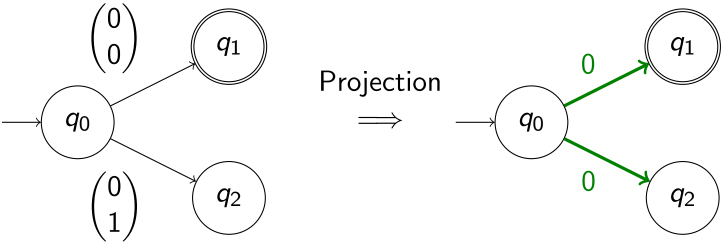
Obtient-on pour autant un automate de taille linéaire en la formule ? Si c'était le cas, on pourrait décider MSO en EXPTIME.

Mais alors quel connecteur logique ne se traduit pas de façon linéaire ?

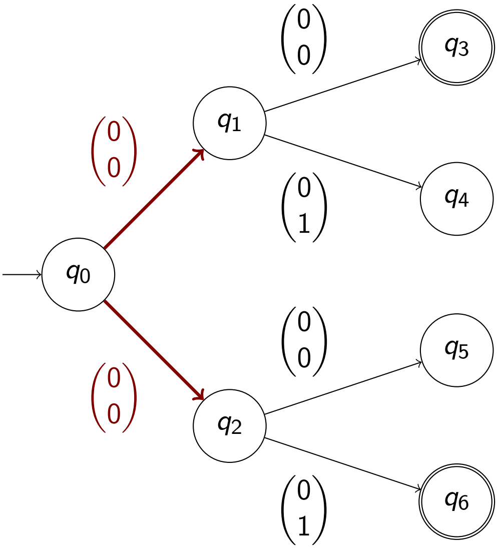
Projections

Les quantificateurs ! Pour un homomorphisme h et un AFA $\mathcal{A}$, on ne sait pas calculer un AFA qui reconnait $h(L(\mathcal{A}))$ de taille polynomiale en celle de $\mathcal{A}$.

Traitement des quantificateurs avec un NFA

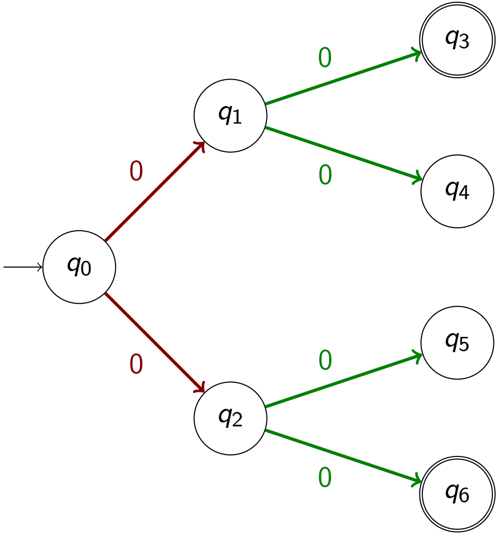


Projection



$$L(\mathcal{A}) = \emptyset$$

Projection



$L = \{0.0\}$

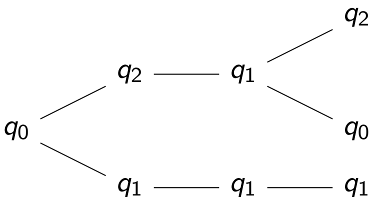
Exécution amnésique

Definition

Soit $\mathcal{A}$ un AFA, $w \in \Sigma^*$, et r une exécution de $\mathcal{A}$ sur w. Celle-ci est *amnésique* si pour tout i et pour tout sommet v, v' de profondeurs i étiquetté par le même état, les sous-arbres partant de v et v' sont identiques.

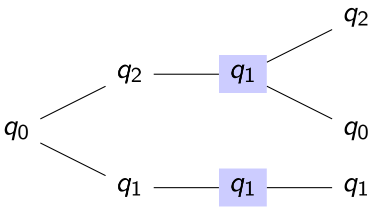
Exemple

δ	a
q_0	$q_1 \wedge q_2$
q_1	$q_1 \vee (q_0 \wedge q_2)$
q_2	q_1



Exemple

δ	a
q_0	$q_1 \wedge q_2$
q_1	$q_1 \vee (q_0 \wedge q_2)$
q_2	q_1



non amnésique

Exemple

δ	a
q_0	$q_1 \wedge q_2$
q_1	$q_1 \vee (q_0 \wedge q_2)$
q_2	q_1

$$\begin{tikzcd}[arrows=dash]
 & q_2 \arrow[r] & q_1 \arrow[r] & q_1 \\
q_0 \arrow[ur] \arrow[dr] & & & \\
 & q_1 \arrow[r] & q_1 \arrow[r] & q_1
\end{tikzcd}$$

amnésique

Théorème d'amnésie

Théorème : Un mot est accepté par un AFA ssi il est accepté par une exécution amnésique.

preuve (informelle) :
étant donné une exécution non amnésique, on peut la remplacer par une exécution amnésique : pour chaque profondeur i et chaque état q, on choisit un sous-arbre débutant à la profondeur i par un q, et on remplace tous les autres sous-arbres débutant à la même profondeur avec la même étiquette q par ce sous-arbre. Cette opération préserve le fait d''être acceptant, et en la répétant pour tout i et q on obtient une exécution amnésique.

fin de la preuve de la non-déterminisation :
l'automate non-déterminisé reconnait un mot w ssi w admet une exécution amnésique dans l'AFA de départ.