INF 302 : Langages & Automates

Chapitre 7 : Automates à États Finis Non-Déterministes avec ϵ-transitions

Yliès Falcone

ylies.falcone@univ-grenoble-alpes.fr — www.ylies.fr

Univ. Grenoble-Alpes, Inria

-

Laboratoire d'Informatique de Grenoble - www.liglab.fr
Équipe de recherche LIG-Inria, Corse - team.inria.fr/corse/

Année Académique 2018 - 2019

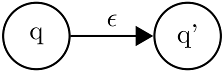
Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Utilisation des ϵ-transitions

On autorise ϵ comme étiquette des transitions ; on parle d'ϵ-transition.



Un mot est accepté si on peut lire le mot jusqu'à un état accepteur :
$\hookrightarrow$ les ϵ-transitions ne "consomment" pas de symbole pendant la lecture du mot d'entrée.

Propriété sous-jacente :

$$\forall w \in \Sigma^* : w \cdot \epsilon = \epsilon \cdot w = w.$$

(ϵ est l'élément neutre de la concaténation entre mots)

Utilisation des ϵ-transitions

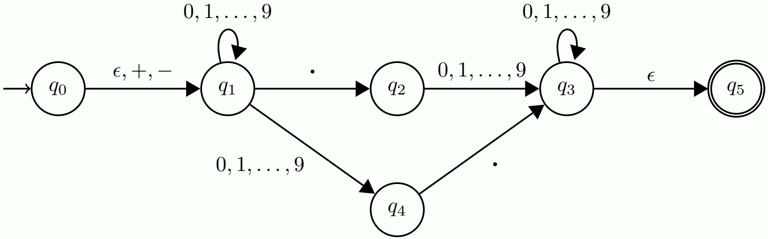
Nombres décimaux

Exemple (Nombres décimaux)

Un nombre écrit en notation décimale consiste en :

- un signe $+$ ou $-$ optionnel,
- un mot de numéros $0, 1, 2, \ldots, 9$,
- un point pour marquer la décimale,
- un mot de numéros $0, 1, 2, \ldots, 9$.

L'un des deux mots de numéros peut être vide, mais ils ne peuvent pas être tous les deux vides.

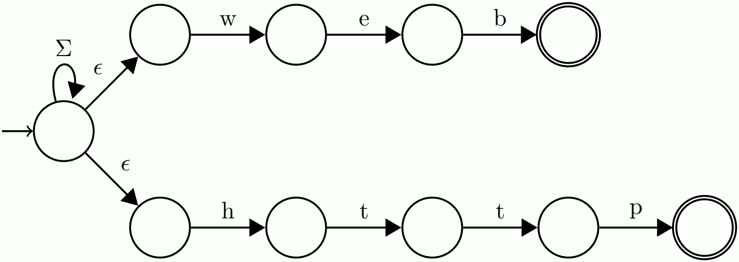


Utilisation des ϵ-transitions

Reconnaissance de mots clés

Exemple (Reconnaissance de mots clés)

- Reconnaissance de deux mots clés : web et http.



- L'ajout d'un mot clé se fait de *manière compositionnelle* :
  - écrire un automate reconnaissant uniquement ce mot clé,
  - ajouter une ϵ-transition depuis l'état initial de l'automate général vers l'état initial de l'automate reconnaissant,
  - l'unique état initial est celui de l'automate général.

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Fermeture de Kleene d'un langage

Soit L un langage sur Σ.

Définition (Fermeture de Kleene)

La Fermeture de Kleene de L, dénotée L^*, est l'ensemble défini inductivement comme l'ensemble généré par les deux règles suivantes :

- $\epsilon \in L^*$, et
- si $u \in L, v \in L^*$, alors $u \cdot v \in L^*$,
- de manière équivalente à la précédente règle : si $u \in L^*, v \in L$, alors $u \cdot v \in L^*$.

On veut montrer que si L est EF alors L^* est EF :

$$\forall L \subseteq \Sigma^* : L \in EF \Rightarrow L^* \in EF$$

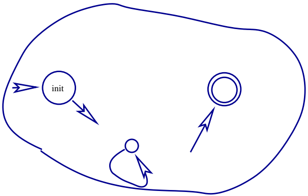
Exemple

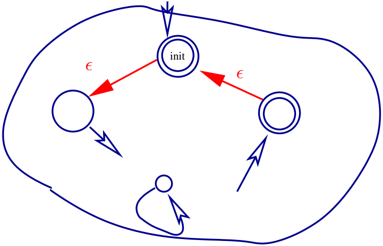
$L = \{ab, u\}$ $L^* = \{u, ab, \varepsilon, uu, abab, uab, abu \}$

Fermeture de Kleene d'un langage - vision automate

Étant donné un automate qui reconnaît L.

Peut-on construire un automate qui reconnaît L^* ?





Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

ANDEF avec ϵ-transitions

Soit Σ un alphabet où le *symbole* $\epsilon \notin \Sigma$.

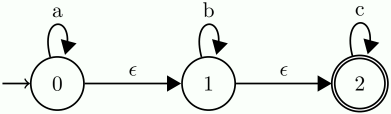
Définition (ANDEF avec ϵ-transitions)

Un *automate à états fini non-déterministe avec ϵ-transitions* (ϵ-ANDEF) est donné par un quintuplet $(Q, \Sigma, q_0, \Delta, F)$ où

- Q est un ensemble fini d'états,
- Σ est l'alphabet de l'automate,
- $q_0 \in Q$ est l'*état initial*,
- $\Delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ est la *relation de transition*,
- $F \subseteq Q$ est l'ensemble des *états terminaux/finaux*.

Exemple (ANDEF avec ϵ-transitions)

Soit $\Sigma = \{a, b, c\}$.



Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Configuration

Soit $A = (Q, \Sigma, q_0, \Delta, F)$ un ϵ-ANDEF.

Définition (Configuration)

Une *configuration* de l'automate A est un couple (q, u) où $q \in Q$ et $u \in \Sigma^*$.

Définition (Relation de dérivation (entre configurations))

On définit la relation $\rightarrow_\Delta$ de *dérivation* entre configurations :

$$(q, a \cdot u) \rightarrow_\Delta (q', u')$$

$$\textit{ssi}$$

$$\big((q, a, q') \in \Delta \text{ et } u' = u\big) \quad \text{ou} \quad \big(a \cdot u = u' \text{ et } (q, \epsilon, q') \in \Delta\big)$$

Notation

- On note $q \xrightarrow{a_1 \cdots a_n}{}^*_\Delta \, q'$ lorsqu'ils existent $q_1, \ldots, q_{n-1}$ tels que :

$$(q, a_1, q_1) \in \Delta, (q_1, a_2, q_2) \in \Delta, \ldots, (q_{n-1}, a_n, q') \in \Delta.$$

- On note $q \longrightarrow^*_\Delta q'$ lorsqu'ils existent $a_1, \ldots, a_n$ tels que $q \xrightarrow{a_1 \cdots a_n}{}^*_\Delta \, q'$.

Exécution

Définition (Exécution)

Une *exécution de l'automate A* est une séquence de configurations $(q_0, u_0) \cdots (q_n, u_n)$ telle que

$$(q_i, u_i) \rightarrow_\Delta (q_{i+1}, u_{i+1}), \text{ pour } i = 0, \ldots, n-1$$

Les notions

- d'acceptation d'un mot, et
- de langage reconnu

sont définies comme dans le cas des ANDEF mutatis mutandis.

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

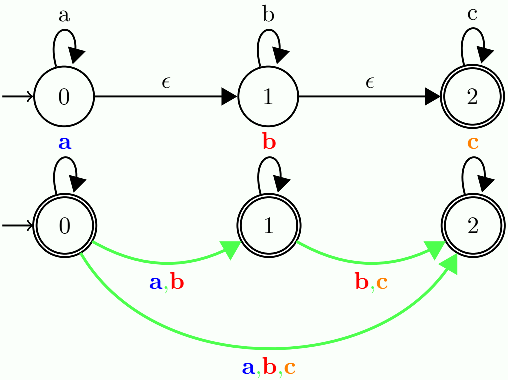
Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Élimination des ϵ-transitions

L'idée sur un exemple

Exemple (ANDEF avec ϵ-transitions)

Soit $\Sigma = \{a, b, c\}$



Élimination des ϵ-transitions : traduction vers ANDEF

Définition

Soit $A = (Q, \Sigma, q_0, \Delta, F)$ un ϵ-ANDEF

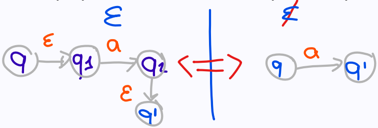
Définition (Élimination des ϵ-transitions)

On construit un ANDEF

$$\epsilon\ell(A) = (Q, \Sigma, q_0, \epsilon\ell(\Delta), \epsilon\ell(F))$$

qui reconnaît $L(A)$ tel que :

- La relation de transition $\epsilon\ell(\Delta)$ est définie par : $(q, a, q') \in \epsilon\ell(\Delta)$ ssi ils existent $q_1, q_2 \in Q$ tels que :
  1. $q \xrightarrow{\epsilon}{}^{*}_{\Delta} q_1$
  2. $(q_1, a, q_2) \in \Delta$
  3. $q_2 \xrightarrow{\epsilon}{}^{*}_{\Delta} q'$



- L'ensemble des états accepteurs $\epsilon\ell(F)$ est défini par :

$$\epsilon\ell(F) = \{q \in Q \mid \exists q' \in F : q \xrightarrow{\epsilon}{}^{*}_{\Delta} q'\}$$

Correction de la procédure d'élimination des ϵ-transitions

Soit $A = (Q, \Sigma, q_0, \Delta, F)$ un ϵ-ANDEF.

Théorème : Correction de la procédure d'élimination des ϵ-transitions

$$L(A) = L(\epsilon\ell(A)).$$

Preuve (par induction)

Pour tout $u, u' \in \Sigma^*$

$$\text{si } (q, u) \xrightarrow{*}_{\epsilon\ell(\Delta)} (q', u') \text{ alors } (q, u) \xrightarrow{*}_{\Delta} (q', u').$$

- Si $\epsilon \in L(A)$ alors $\epsilon \in L(\epsilon\ell(A))$
- Pour tout $u \in \Sigma^*$ avec $|u| > 0$,

$$\text{si } (q, u) \xrightarrow{*}_{\Delta} (q', u') \text{ alors } (q, u) \xrightarrow{*}_{\epsilon\ell(\Delta)} (q', u')$$

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Fermeture par ϵ (ϵ-fermeture)

Définition

L'ϵ-fermeture d'un état q consiste à regrouper tous les états qu'on peut atteindre en suivant toutes les transitions sortantes de q et étiquetées par ϵ

Définition (ϵ-Fermeture d'un état)

Soit $q \in Q$ un état, on définit $ECLOSE(q)$ de façon récursive :

- *Case de base :* $q \in ECLOSE(q)$
- *Induction :* Si $p \in ECLOSE(q)$, et s'il existe une transition de p vers $r \in Q$ étiquetée par ϵ, alors $r \in ECLOSE(q)$

De manière équivalente : $ECLOSE(q) = \delta^*(q, \epsilon)$

Définition (ϵ-Fermeture d'un ensemble d'états)

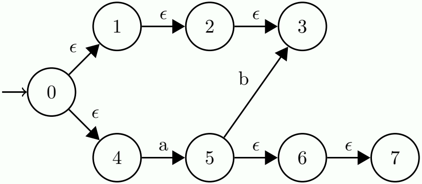
Pour $S \subseteq Q$:

$$ECLOSE(S) = \bigcup_{q \in S} ECLOSE(q)$$

Fermeture par ϵ (ϵ-fermeture)

Exemple

Exemple (ϵ-Fermeture)



- $ECLOSE(0) = \{0, 1, 2, 3, 4\}$
- $ECLOSE(3) = \{3\}$
- $ECLOSE(5) = \{5, 6, 7\}$

Relation de transition étendue

On définit la relation de transition étendue $\hat{\delta}$ qui permet de lire des entrées contenant possiblement des symboles différents de ϵ.

Intuitivement, $\hat{\delta}(q, w)$ est l'ensemble d'états atteints en suivant un chemin dont les étiquettes concaténées forment w (et ϵ ne "contribue" pas à w).

Définition (Relation de transition étendue)

Étant donnés $q \in Q$ et $w \in (\Sigma \cup \{\epsilon\})^*$:

- *Cas de base :* $\hat{\delta}(q, \epsilon) = ECLOSE(q)$
- *Induction :* Pour $w = x \cdot a$, avec $a \in \Sigma$, $\hat{\delta}(q, x \cdot a)$ est défini par :
  - soit $\{p_1, p_2, \ldots, p_k\} = \hat{\delta}(q, x)$,
  - soit $\{r_1, r_2, \ldots, r_m\} = \bigcup_{i=1}^{k} \delta(p_i, a)$,
  - alors $\hat{\delta}(q, w) = ECLOSE(\{r_1, r_2, \ldots, r_m\})$.

Élimination des ϵ-transitions et déterminisation “à la volée"

Traduction vers ADEF - définition du déterminisé

Soit $A = (Q, \Sigma, q_0, \Delta, F)$ un ϵ-ANDEF

Définition (Déterminisation et élimination des ϵ-transitions, à la volée)

Le *déterminisé* de A est l'ADEF

$$(Q_D, \Sigma, q_D, \delta, F_D)$$

tel que :

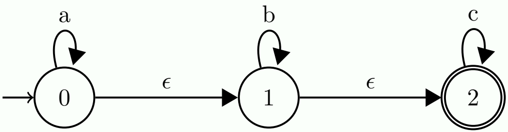
- $Q_D = \mathcal{P}(Q)$
- $q_D = ECLOSE(q_0)$
- δ est définie comme suit : pour tout $S \in Q_D, a \in \Sigma$:
  - soit $\{p_1, p_2, \ldots, p_k\} = S$,
  - soit $\{r_1, r_2, \ldots, r_m\} = \bigcup_{i=1}^{k} \Delta(p_i, a)$,
  - alors $\delta(S, a) = ECLOSE(\{r_1, r_2, \ldots, r_m\})$,
- $F_D = \{S \in \mathcal{P}(Q) \mid S \cap F \neq \emptyset\}$.

Remarque Chaque état de l'automate déterminisé correspond à un ensemble d'états de l'automate non-déterministe avec ϵ-transitions qui est ϵ-fermé. $\square$

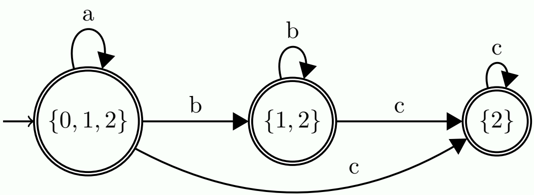
Élimination des ϵ-transitions et déterminisation “à la volée"

Traduction vers ADEF : exemple

Exemple (Élimination des ϵ-transitions - Traduction vers ADEF)



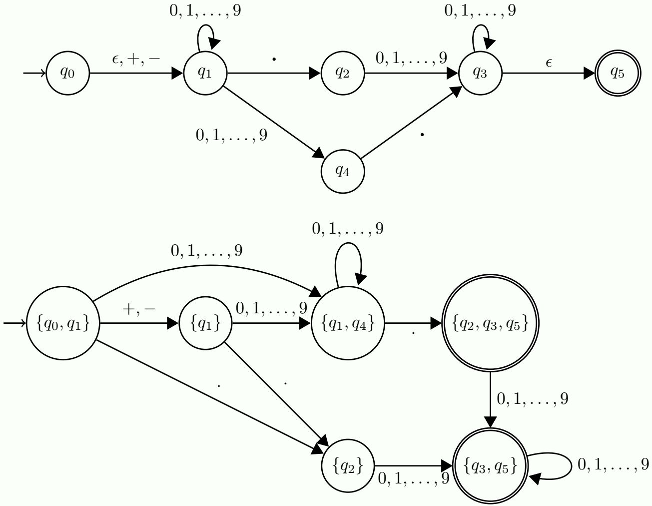
On combine élimination des ϵ-transitions et déterminisation.
On fait les deux opérations “à la volée".



Élimination des ϵ-transitions et déterminisation “à la volée"

Traduction vers ADEF : un autre exemple

Exemple (Élimination des ϵ-transitions - Traduction vers ADEF)

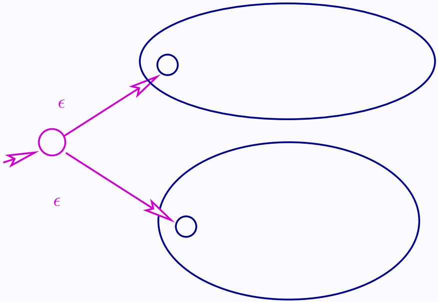


Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

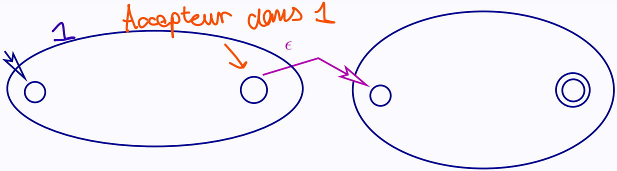
Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Fermeture des langages à états par union et concaténation

Union



Concaténation



Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Opération miroir

Le miroir d'un mot est le mot écrit en *lisant de droite à gauche*.

Définition (Opération miroir – mot et langage)

- Pour $w = a_1 \cdot a_2 \cdots a_n$, le *miroir* de w est le mot dénoté w^R et défini par :

$$w^R = a_n \cdot a_{n-1} \cdots a_1$$

- Pour $L \subseteq \Sigma^*$, le *miroir* de L est le langage, dénoté L^R, des mots miroirs de L :

$$L^R = \{w^R \mid w \in L\}$$

Exemple (Opération miroir)

Pour $L = \{001, 10, 111\}$, on a $L^R = \{100, 01, 111\}$.

Fermeture des langages à états par opération miroir

Fermeture de EF par l'opération miroir

- Si $L \subseteq \Sigma^*$ est un langage à états, alors ainsi est L^R.
- Donc EF est fermé par l'opération miroir.

Preuve informelle basée sur les automates

Étant donnés un langage L à états et son automate reconnaisseur A :

1. Inverser toutes les transitions de A.
2. Faire de l'état initial de A l'unique état accepteur.
3. Créer un nouvel état initial q_0.
4. Ajouter une transition étiquettée par ϵ depuis q_0 vers chaque état accepteur de l'automate initial.

(La preuve est laissée sous forme d'exercice en TD.)

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Rappel : morphismes (de groupes)

Soient $(G, \bullet)$ et $(G', *)$ deux groupes dont les éléments neutres sont e_G et $e_{G'}$, respectivement.

Définition (Morphisme)

Une application $f : G \to G'$ est un morphisme de groupes si

$$\forall x, y \in G : f(x \bullet y) = f(x) * f(y)$$

Exemple (Morphisme)

L'application $f : (Z, +) \to (R, \times)$ définie par $f(n) = 2^n$ est un morphisme de groupes.

Morphisme sur les mots

Soit Σ et Σ' deux alphabets.

Définition (Morphisme de mots)

Une application $h : \Sigma \to \Sigma'^*$ induit un morphisme $\hat{h}$, de $\Sigma^* \to \Sigma'^*$ de la manière suivante :

- $\hat{h}(\epsilon) = \epsilon$, et
- $\hat{h}(u \cdot a) = \hat{h}(u) \cdot h(a)$.

Remarque On montre que $\hat{h}$ est un morphisme en montrant $\forall x, y \in \Sigma^* = \hat{h}(x \cdot y) = \hat{h}(x) \cdot \hat{h}(y)$ par induction sur y ou récurrence sur la longueur de y. $\square$

Exemple (Morphisme de mots)

Considérons $\Sigma = \{a, b\}$, $\Sigma' = \{0, 1\}$ et l'application $h : \Sigma \to \Sigma'^*$ telle que $h(a) = 0$ et $h(b) = 1 \cdot 1$.
L'application h induit bien un morphisme $\hat{h} : \Sigma^* \to \Sigma'^*$. En effet, on a par exemple $\hat{h}(b \cdot a \cdot a) = 1 \cdot 1 \cdot 0 \cdot 0 = \hat{h}(b) \cdot \hat{h}(a \cdot a)$.

À partir de maintenant, on écrit h au lieu de $\hat{h}$.

Fermeture des langages à états par morphisme

Soit h un morphisme.

Théorème : Fermeture de EF par morphisme

Si $L \subseteq \Sigma^*$ est un langage à états alors ainsi est son image $h(L)$ par h définie par

$$h(L) = \{h(u) \mid u \in L\} .$$

Donc EF est fermé par morphisme.

Exemple (Fermeture de EF par morphisme)

Considérons $\Sigma = \{a, b\}$, $\Sigma' = \{0, 1\}$ et le morphisme $\hat{h}$ (noté h ci-dessous) comme décrit précédemment et induit par l'application $h : \Sigma \to \Sigma'^*$ telle que $h(a) = 0$ et $h(b) = 1 \cdot 1$.

- Le langage $L_1 \subseteq \Sigma^*$ contenant l'ensemble des mots avec un nombre impair de a est un langage à états.
- Le langage $h(L_1) \subseteq \Sigma'^*$ contenant l'ensemble des mots avec un nombre impair de 0 et un nombre pair de 1 est un langage à états.

Fermeture de EF par morphisme inverse

Soit h un morphisme et h^{-1} le morphisme inverse (application inverse).

Théorème : Fermeture de EF par morphisme inverse

Si $L \subseteq \Sigma'^*$ est un langage à états alors ainsi est son image $h^{-1}(L)$ par h^{-1} définie par

$$h^{-1}(L) = \left\{ u \in \Sigma^* \mid \exists u' \in L : h(u) = u' \right\}.$$

Donc EF est fermé par morphisme inverse.

Exemple (Fermeture de EF par morphisme inverse)

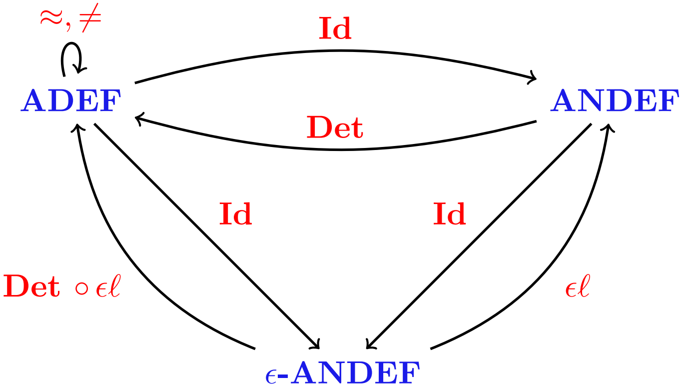
Considérons $\Sigma = \{a, b, c, d\}$, $\Sigma' = \{0, 1, 2\}$ et le morphisme $\hat{h}$ (noté h ci-dessous) induit par l'application $h : \Sigma \rightarrow \Sigma'^*$ telle que $h(a) = 0$, $h(b) = 1$, $h(c) = \epsilon$ et $h(d) = 2$.

- Le langage $L_1 \subseteq \Sigma'^*$ contenant l'ensemble des mots avec un nombre pair de 0 et pas de 2 est un langage à états.
- Le langage $h^{-1}(L_1) \subseteq \Sigma'^*$ contenant l'ensemble des mots avec
  - un nombre pair de 0,
  - pas de d, et
  - des b et des c de manière non-contrainte

  est un langage à états.

Plan Chap. 7 - Automates à États Finis Non-Déterministes avec ϵ-transitions

Résumé 1 : transformations entre automates



Résumé 2 : fermeture de la classe des langages à états et problèmes / procédures de décision

Propriétés de fermeture

Les langages d'états finis sont fermés par les opérations suivantes :

1. union, intersection,
2. complément,
3. concaténation,
4. l'étoile de Kleene (L^*),
5. opération miroir,
6. morphisme, morphisme inverse.

Nous avons associé ces opérations à des transformations d'automates.

Problèmes et procédures de décision

Les problèmes de décisions suivants sont décidables :

1. accessibilité,
2. co-accessibilité,
3. langage vide,
4. langage infini,
5. inclusion de langages,
6. égalité de langages.

Nous avons donné une procédure de décision pour chacun d'entre eux.