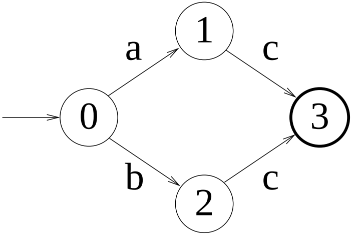
Théorie des langages

Alexis Nasr

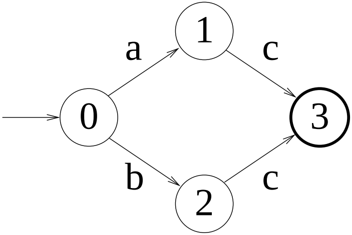
Nombre d'états d'un automate

- Etant donné un automate déterministe A qui reconnaît le langage L,
- est il possible de reconnaître L avec un automate déterministe A' qui possède moins d'état que A ?
- Idée générale : étant donné deux états p et q, si tous les mots que l'on peut reconnaître à partir de p peuvent aussi l'être à partir de q, alors on pourrait regrouper p et q au sein d'un même état.

Exemple

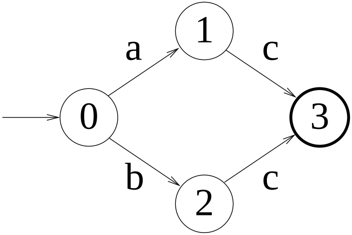


Exemple



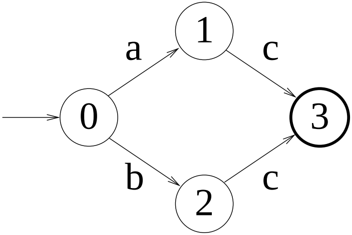
- Tous les mots qui peuvent être reconnus à partir de 1 peuvent aussi l'être à partir de 2.

Exemple

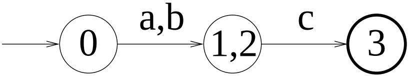


- Tous les mots qui peuvent être reconnus à partir de 1 peuvent aussi l'être à partir de 2.
- On regroupe 1 et 2 au sein de l'état $1, 2$

Exemple



- Tous les mots qui peuvent être reconnus à partir de 1 peuvent aussi l'être à partir de 2.
- On regroupe 1 et 2 au sein de l'état $1, 2$



Automate minimal

- Soit $L \subseteq \Sigma^*$ un langage reconnaissable. L'automate minimal de L est, parmi les automates déterministes complets reconnaissant L, celui qui possède le moins d'états.
- L'automate minimal de L est unique au renommage des états près.

Automate minimal

- L'automate minimal d'un langage reconnaissable est son automate des résiduels

Automate minimal

- L'automate minimal d'un langage reconnaissable est son automate des résiduels

Preuve :

- Soient L un langage reconnaissable et $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ son automate des résiduels.
- Alors $Q = \{L/w, w \in \Sigma^*\}$ et $|Q|$ est exactement le nombre de résiduels de L.
- Or on sait que $|Q|$ minore le nombre d'états de tout automate déterministe complet qui reconnaît L
- A est donc l'automate minimal de L.

Equivalence de deux automates

- Etant donné deux automates A et A' avec $L = L(A)$ et $L' = L(A')$
- Comment savoir si $L = L'$?
- L et L' étant reconnaissables, on sait qu'il existe un seul automate minimal M tel que $L = L(M)$ et un seul automate M' tel que $L' = L(M')$
- Si l'on sait construire M à partir de A et M' à partir de A', alors, en comparant M et M' on pourra répondre à la question.
- Il existe un algorithme permettant de construire l'automate minimal d'un automate quelconque.

Relation binaire

Etant donné deux ensembles X et Y

- Une relation binaire entre X et Y est un sous-ensemble du produit cartésien $X \times Y$, en d'autres termes, une collection de couples dont la première composante est dans X et la seconde dans Y.
- Les composantes d'un couple appartenant à une relation R sont dits en relation par R.
- Dans le cas particulier où $X = Y$ on dit que R est une relation binaire définie sur X ou dans X.

Relation binaire

Etant donné deux ensembles X et Y

- Une relation binaire entre X et Y est un sous-ensemble du produit cartésien $X \times Y$, en d'autres termes, une collection de couples dont la première composante est dans X et la seconde dans Y.
- Les composantes d'un couple appartenant à une relation R sont dits en relation par R.
- Dans le cas particulier où $X = Y$ on dit que R est une relation binaire définie sur X ou dans X.
  - Exemple 1 : $X = \{aa, aba, bab, bb\}$, R_1 est la relation binaire sur X "commence par le même symbole que". On a :
    $R_1 = \{(aa, aba), (aba, aa), (aa, aa), (aba, aba), (bab, bb),$
    $(bb, bab), (bb, bb), (bb, bab)\}$

Relation binaire

Etant donné deux ensembles X et Y

- Une relation binaire entre X et Y est un sous-ensemble du produit cartésien $X \times Y$, en d'autres termes, une collection de couples dont la première composante est dans X et la seconde dans Y.
- Les composantes d'un couple appartenant à une relation R sont dits en relation par R.
- Dans le cas particulier où $X = Y$ on dit que R est une relation binaire définie sur X ou dans X.
  - Exemple 1 : $X = \{aa, aba, bab, bb\}$, R_1 est la relation binaire sur X "commence par le même symbole que". On a :
    $R_1 = \{(aa, aba), (aba, aa), (aa, aa), (aba, aba), (bab, bb),$
    $(bb, bab), (bb, bb), (bb, bab)\}$
  - Exemple 2 : R_2 est la relation binaire sur X "est de longueur strictement supérieure à"
    $R_2 = \{(aba, aa), (aba, bb), (bab, aa), (bab, bb)\}$

Relation d'équivalence

Etant donné un ensemble X

- Une relation d'équivalence R sur X est une relation binaire sur X qui est :
  - réflexive : $\forall x \in X,\ xRx$
  - symétrique : $\forall x, y \in X,\ xRy \Rightarrow yRx$
  - transitive : $\forall x, y, z \in X,\ xRy$ et $yRz \Rightarrow xRz$
- R_1 est une relation d'équivalence
- R_2 n'en est pas une

Classe d'équivalence

Etant donné un ensemble X et une relation d'équivalence R

- La classe d'équivalence de $x \in X$, notée x/R, est l'ensemble :

$$x/R = \{y \in X \mid xRy\}$$

Classe d’équivalence

Etant donné un ensemble X et une relation d’équivalence R

- La classe d’équivalence de $x \in X$, notée x/R, est l’ensemble :

$$x/R = \{y \in X \mid xRy\}$$

  - Exemple : $aa/R_1 = \{aa, aba\}$

Ensemble quotient

Etant donné un ensemble X et une relation d'équivalence R

- L'ensemble quotient de X par la relation d'équivalence R, noté X/R, est l'ensemble des classes d'équivalence de X suivant R :

$$X/R = \{x/R | x \in X\}$$

Ensemble quotient

Etant donné un ensemble X et une relation d'équivalence R

- L'ensemble quotient de X par la relation d'équivalence R, noté X/R, est l'ensemble des classes d'équivalence de X suivant R :

$$X/R = \{x/R | x \in X\}$$

  - Exemple : $X/R_1 = \{\{aa, aba\}, \{bb, bab\}\}$

Congruence

- Deux entiers a et b sont dits congruents modulo n ($a \equiv b(n)$), où n est un entier supérieur ou égal à 2, si l'une des conditions équivalentes suivantes est vérifiée :
  - leur différence est divisible par n ; (il existe un entier k tel que $a - b = kn$)
  - le reste de la division euclidienne de a par n est égal à celui de la division de b par n ;
- Exemple : $10 \equiv 19(9)$

Congruence

- La congruence modulo n est :
  - Réflexive :
    $$a \equiv a(n)$$
  - Symétrique :
    $$a \equiv b(n) \Leftrightarrow b \equiv a(n)$$
  - Transitive :
    $$\text{si } a \equiv b(n) \text{ et } b \equiv c(n) \text{ alors } a \equiv c(n)$$
- Il s'agit donc d'une relation d'équivalence.

Congruence sur un automate

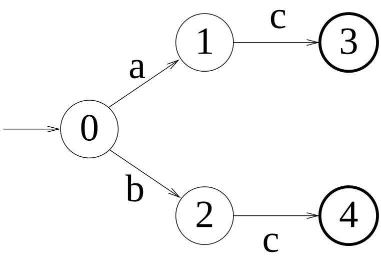
- Soit $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe.
- Une relation d'équivalence $\sim$ sur Q est une congruence sur A si :
  - elle est compatible avec δ :

$$(p \sim q) \Rightarrow \forall a \in \Sigma \ : \ \delta(p, a) \sim \delta(q, a)$$

  - elle sature F :

$$(p \sim q) \Rightarrow (p \in F \Leftrightarrow q \in F)$$

Exemple



- 3 est congru à 4
- 1 est congru à 2

Automate quotient

- Soit $A = \langle Q, \Sigma, \delta, i, F \rangle$ un automate déterministe et $\sim$ une congruence sur A.
- L'automate quotient de A par $\sim$, noté $A_\sim$ est l'automate obtenu en fusionnant les états appartenant à la même classe d'équivalence.
- Il est défini par $A_\sim = \langle Q_\sim, \Sigma, \delta_\sim, i_\sim, F_\sim \rangle$ où :
  - $Q_\sim = Q/\sim = \{q/\sim, q \in Q\}$
  - $\delta_\sim(q/\sim, a) = \delta(q, a)/\sim, \forall q \in Q, \forall a \in \Sigma$
  - $i_\sim = i/\sim$
  - $F_\sim = F/\sim$

Langage de l'automate quotient

Pour toute congruence $\sim$ sur un automate déterministe A, on a :

$$L(A_{\sim}) = L(A)$$

Langage de l'automate quotient

Pour toute congruence $\sim$ sur un automate déterministe A, on a :

$$L(A_\sim) = L(A)$$

Preuve

- par définition de $\delta_\sim$, on a :

  $$\delta_\sim(p/\sim, a) = q/\sim \Leftrightarrow \delta(p, a) = q$$

  pour tout $p/\sim, q/\sim \in Q_\sim$
  et tout $a \in \Sigma$.
- Il s'ensuit que $(i_\sim, u) \overset{*}{\vdash} (q/\sim, \varepsilon) \Leftrightarrow (i, u) \overset{*}{\vdash} (q, \varepsilon)$
- de plus, $q/\sim \in F_\sim \Leftrightarrow q \in F$
- par conséquent $u \in L(A_\sim) \Leftrightarrow u \in L(A)$

Congruence de Nerode

- Soit $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe
- La congruence de Nerode de A est la relation d'équivalence $\approx$ définie sur Q par :

$$p \approx q \text{ si } L_p = L_q$$

Automate minimal

- Soit A un automate déterministe complet reconnaissant un langage L. L'automate minimal M de L est égal à l'automate quotient $A_{\approx}$, où $\approx$ est la congruence de Nérode de A.

Automate minimal

- Soit A un automate déterministe complet reconnaissant un langage L. L'automate minimal M de L est égal à l'automate quotient $A_{\approx}$, où $\approx$ est la congruence de Nérode de A.

Preuve

- puisque $\approx$ est une congruence sur A, l'automate $A_{\approx}$ reconnaît L.
- reste à montrer que l'ensemble Q des états de $A_{\approx}$ est de cardinal minimal.
- Or $Q_{\approx} = \{q/\approx, q \in Q\}$ et cet ensemble est clairement équipotent à $\{L_q, q \in Q\}$ qui, nous l'avons vu, coïncide avec $\{L/u, u \in \Sigma^*\}$
- Par conséquent $|Q_{\approx}|$ est égal au nombre de résiduels de L, qui minimise le nombre d'états de tout automate déterministe qui reconnaît L.

Minimisation

- Soit $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate déterministe qui reconnaît L.
- On peut construire l'automate minimal M de L
  - en identifiant les classes d'équivalences de la congruence de Nerode de A,
  - puis en fusionnant les états équivalents de A.

Construction des classes d'équivalence

- On note $\Sigma^{\leq n} = \{u \in \Sigma^*, |u| \leq n\}$
- On définit la relation $\approx_n$:

$$p \approx_n q \; ssi \; \forall u \in \Sigma^{\leq n} : u \in L_p \Leftrightarrow u \in L_q$$

Construction des classes d'équivalence

- On exprime $\approx_{n+1}$ en fonction de $\approx_n$

Construction des classes d'équivalence

- On exprime $\approx_{n+1}$ en fonction de $\approx_n$
- $v \in \Sigma^{\leq n+1}$ *ssi* $v \in \Sigma^{\leq n}$ ou bien $v = au$ avec $a \in \Sigma$ et $u \in \Sigma^{\leq n}$

Construction des classes d'équivalence

- On exprime $\approx_{n+1}$ en fonction de $\approx_n$
- $v \in \Sigma^{\leq n+1}$ *ssi* $v \in \Sigma^{\leq n}$ ou bien $v = au$ avec $a \in \Sigma$ et $u \in \Sigma^{\leq n}$
- Par conséquent $p \approx_{n+1} q$ *ssi*
  $\forall a \in \Sigma, \forall u \in \Sigma^{\leq n} : (u \in L_p \Leftrightarrow u \in L_q)$ et $(au \in L_p \Leftrightarrow au \in L_q)$

Construction des classes d'équivalence

- On exprime $\approx_{n+1}$ en fonction de $\approx_n$
- $v \in \Sigma^{\leq n+1}$ *ssi* $v \in \Sigma^{\leq n}$ ou bien $v = au$ avec $a \in \Sigma$ et $u \in \Sigma^{\leq n}$
- Par conséquent $p \approx_{n+1} q$ *ssi*
  $\forall a \in \Sigma, \forall u \in \Sigma^{\leq n} : (u \in L_p \Leftrightarrow u \in L_q)$ et $(au \in L_p \Leftrightarrow au \in L_q)$
- Or $\forall p \in Q : au \in L_p \Leftrightarrow u \in L_{\delta(p,a)}$

Construction des classes d'équivalence

- On exprime $\approx_{n+1}$ en fonction de $\approx_n$
- $v \in \Sigma^{\leq n+1}$ *ssi* $v \in \Sigma^{\leq n}$ ou bien $v = au$ avec $a \in \Sigma$ et $u \in \Sigma^{\leq n}$
- Par conséquent $p \approx_{n+1} q$ *ssi*
  $\forall a \in \Sigma, \forall u \in \Sigma^{\leq n} : (u \in L_p \Leftrightarrow u \in L_q)$ et $(au \in L_p \Leftrightarrow au \in L_q)$
- Or $\forall p \in Q : au \in L_p \Leftrightarrow u \in L_{\delta(p,a)}$
- Donc $p \approx_{n+1} q$ *ssi*
  $\forall a \in \Sigma$ et $u \in \Sigma^{\leq n} : (u \in L_p \Leftrightarrow u \in L_q)$ et $(u \in L_{\delta(p,a)} \Leftrightarrow u \in L_{\delta(q,a)})$

Construction des classes d'équivalence

- On exprime $\approx_{n+1}$ en fonction de $\approx_n$
- $v \in \Sigma^{\leq n+1}$ *ssi* $v \in \Sigma^{\leq n}$ ou bien $v = au$ avec $a \in \Sigma$ et $u \in \Sigma^{\leq n}$
- Par conséquent $p \approx_{n+1} q$ *ssi*
  $\forall a \in \Sigma, \forall u \in \Sigma^{\leq n} : (u \in L_p \Leftrightarrow u \in L_q)$ et $(au \in L_p \Leftrightarrow au \in L_q)$
- Or $\forall p \in Q : au \in L_p \Leftrightarrow u \in L_{\delta(p,a)}$
- Donc $p \approx_{n+1} q$ *ssi*
  $\forall a \in \Sigma$ et $u \in \Sigma^{\leq n} : (u \in L_p \Leftrightarrow u \in L_q)$ et $(u \in L_{\delta(p,a)} \Leftrightarrow u \in L_{\delta(q,a)})$
- Qui s'écrit aussi :

$$p \approx_{n+1} q \text{ } ssi \text{ } (p \approx_n q \text{ et } \forall a \in \Sigma : \delta(p,a) \approx_n \delta(q,a))$$

Construction des classes d'équivalence

- On construit les classes d'équivalences de $\approx_0, \approx_1, \ldots, \approx$ pas à pas :
- on commence par $\approx_0$
- puis, $\forall n > 0$, $\approx_n$ en fonction de $\approx_{n-1}$
- jusqu'à ce que $\approx_n = \approx_{n-1}$

Calcul des classes d'équivalence de $\approx_0$

- $p \approx_0 q$ *ssi* $\forall u \in \Sigma^{\leq 0} : u \in L_p \Leftrightarrow u \in L_q$
- en d'autres termes : $p \approx_0 q$ *ssi* $\varepsilon \in L_p \Leftrightarrow \varepsilon \in L_q$
- Or $\varepsilon \in L_p \Leftrightarrow p \in F$ où F est l'ensemble des états d'acceptation.
- D'où, $p \approx_0 q$ *ssi* $p, q \in F$ ou $p, q \in Q - F$
- Ainsi, $\approx_0$ définit deux classes d'équivalence exactement : F et $Q - F$

Calcul des classes d'équivalence de $\approx_1$

- $p \approx_1 q$ si $p \approx_0 q$ et si $\forall a \in \Sigma, \delta(p, a) \approx_1 \delta(q, a)$
- Autrement dit, les classes de $\approx_1$ sont obtenues en séparant, dans chaque classe de $\approx_0$, les états qui sont envoyés par un même symbole sur des classes de $\approx_0$ différentes.
- On calcule de même $\approx_2$ en fonction de $\approx_1$ et, de proche en proche, toutes les relations $\approx_n$.
- Il existe certainement un entier n pour lequel $\approx_n$ et $\approx$ coïncident.

Exemple 1

		a	b
$\rightarrow$	1	2	4
	2	3	6
$\leftarrow$	3	3	3
	4	5	2
	5	3	6
$\leftarrow$	6	6	4

$\approx_0$	1,2,4,5	3,6

Exemple 1

		a	b
$\rightarrow$	1	2	4
	2	3	6
$\leftarrow$	3	3	3
	4	5	2
	5	3	6
$\leftarrow$	6	6	4

<table>
<tr><td>$\approx_0$</td><td colspan="2">1,2,4,5</td><td colspan="2">3,6</td></tr>
<tr><td>$\approx_1$</td><td>1,4</td><td>2,5</td><td>3</td><td>6</td></tr>
</table>

Exemple 1

		a	b
$\rightarrow$	1	2	4
	2	3	6
$\leftarrow$	3	3	3
	4	5	2
	5	3	6
$\leftarrow$	6	6	4

<table>
<tr><td>$\approx_0$</td><td colspan="3">1,2,4,5</td><td colspan="2">3,6</td></tr>
<tr><td>$\approx_1$</td><td colspan="2">1,4</td><td>2,5</td><td>3</td><td>6</td></tr>
<tr><td>$\approx_2$</td><td>1</td><td>4</td><td>2,5</td><td>3</td><td>6</td></tr>
</table>

Exemple 2

		a	b
$\rightarrow$	1	2	3
	2	2	4
	3	3	5
$\leftarrow$	4	4	5
$\leftarrow$	5	5	4

$\approx_0$	1,2,3	4,5

Exemple 2

		a	b
$\rightarrow$	1	2	3
	2	2	4
	3	3	5
$\leftarrow$	4	4	5
$\leftarrow$	5	5	4

<table>
<tr><td>$\approx_0$</td><td colspan="2">1,2,3</td><td>4,5</td></tr>
<tr><td>$\approx_1$</td><td>1</td><td>2,3</td><td>4,5</td></tr>
</table>