2. Autòmats finits

Teoria de la Computació

FIB

Antoni Lozano

Q1 2024–2025

Autòmats finits

1. Concepte
2. Exemples
3. Definició formal
4. Llenguatge reconegut
5. Tancament per operacions 1
6. Indeterminisme
7. Tancament per operacions 2

Autòmats finits

1. Concepte
2. Exemples
3. Definició formal
4. Llenguatge reconegut
5. Tancament per operacions 1
6. Indeterminisme
7. Tancament per operacions 2

Teoria d'autòmats

La teoria d'autòmats és l'estudi de les màquines abstractes (models abstractes de càlcul) definides amb

- un mecanisme format per estats i transicions (control finit)
- una possible memòria externa (en forma de pila o cinta)

L'autòmat finit (FA, de *finite automaton*) és el model abstracte de càlcul més senzill de la teoria d'autòmats i no conté memòria addicional a la donada pels estats.

Teoria d'autòmats

La teoria d'autòmats és l'estudi de les màquines abstractes (models abstractes de càlcul) definides amb

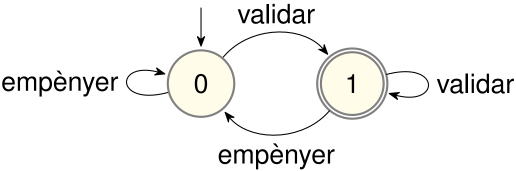
- un mecanisme format per estats i transicions (control finit)
- una possible memòria externa (en forma de pila o cinta)

L'autòmat finit (FA, de *finite automaton*) és el model abstracte de càlcul més senzill de la teoria d'autòmats i no conté memòria addicional a la donada pels estats.

Els autòmats finits són models d'ordinadors que només necessiten una quantitat fixa de memòria, que es pot representar amb els estats.

Molts aparells que ens envolten necessiten una quantitat fixa (i sovint molt petita) de memòria. Un torniquet de metro només necessita 2 estats:

0 (bloquejat) i 1 (desbloquejat).

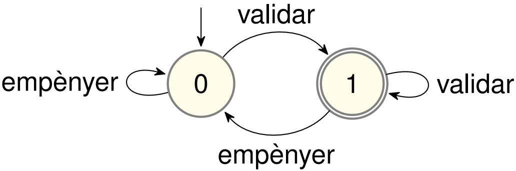


El diagrama anterior s'anomena diagrama d'estats. Té un estat inicial (indicat amb una fletxa sense origen) i un conjunt d'estats acceptadors (indicats amb doble cercle o una creu).

Els autòmats finits són models d'ordinadors que només necessiten una quantitat fixa de memòria, que es pot representar amb els estats.

Molts aparells que ens envolten necessiten una quantitat fixa (i sovint molt petita) de memòria. Un torniquet de metro només necessita 2 estats:

0 (bloquejat) i 1 (desbloquejat).



El pas d'un estat a un altre donat per una acció concreta s'anomena transició. Per exemple, si en l'estat 0 es fa **validar**, passem a l'estat 1 i diem que hi ha una transició de 0 a 1.

Una segona representació és la taula de transicions:

	validar	**empènyer**	
0	1	0	
1	1	0	+

La taula de transicions pot ser una descripció completa de l'autòmat:

- L'estat inicial és el primer de la taula
- Els acceptadors s'indiquen amb una creu a la dreta
  (un signe + en RACSO)

Una segona representació és la taula de transicions:

	validar	empènyer	
0	1	0	
1	1	0	+

La taula de transicions pot ser una descripció completa de l'autòmat:

- L'estat inicial és el primer de la taula
- Els acceptadors s'indiquen amb una creu a la dreta
  (un signe + en RACSO)

Suposem que ara volem donar al torniquet memòria per una persona més. És a dir, si algú valida abans que passi la persona que el precedeix, el torniquet ha de permetre el pas dues vegades seguides.

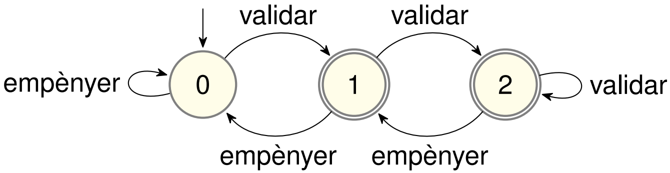
(El torniquet permet el pas si i només si s'empeny en un estat acceptador)

L'autòmat finit que volem és:

Suposem que ara volem donar al torniquet memòria per una persona més. És a dir, si algú valida abans que passi la persona que el precedeix, el torniquet ha de permetre el pas dues vegades seguides.

(El torniquet permet el pas si i només si s'empeny en un estat acceptador)

L'autòmat finit que volem és:



Autòmats finits

1 Concepte
2 Exemples
3 Definició formal
4 Llenguatge reconegut
5 Tancament per operacions 1
6 Indeterminisme
7 Tancament per operacions 2

En un context formal, l'acció que permet fer una transició és la lectura d'un símbol d'un alfabet.

Un mot és acceptat si, partint de l'estat inicial, la lectura dels seus símbols permet arribar a un estat acceptador a través de transicions.

Exemple: autòmat finit sobre alfabet $\{0, 1\}$

El llenguatge reconegut per un autòmat finit és el conjunt dels mots acceptats.

Qüestió

Quin és el llenguatge reconegut per l'autòmat finit anterior?

En un context formal, l'acció que permet fer una transició és la lectura d'un símbol d'un alfabet.

Un mot és acceptat si, partint de l'estat inicial, la lectura dels seus símbols permet arribar a un estat acceptador a través de transicions.

Exemple: autòmat finit sobre alfabet $\{0, 1\}$

El llenguatge reconegut per un autòmat finit és el conjunt dels mots acceptats.

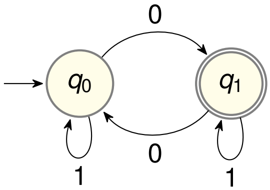
Qüestió

Quin és el llenguatge reconegut per l'autòmat finit anterior?

En un context formal, l'acció que permet fer una transició és la lectura d'un símbol d'un alfabet.

Un mot és acceptat si, partint de l'estat inicial, la lectura dels seus símbols permet arribar a un estat acceptador a través de transicions.

Exemple: autòmat finit sobre alfabet $\{0, 1\}$



El llenguatge reconegut per un autòmat finit és el conjunt dels mots acceptats.

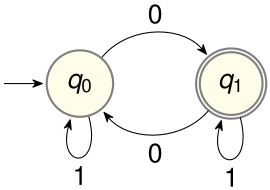
Qüestió

Quin és el llenguatge reconegut per l'autòmat finit anterior?

En un context formal, l'acció que permet fer una transició és la lectura d'un símbol d'un alfabet.

Un mot és acceptat si, partint de l'estat inicial, la lectura dels seus símbols permet arribar a un estat acceptador a través de transicions.

Exemple: autòmat finit sobre alfabet $\{0, 1\}$



El llenguatge reconegut per un autòmat finit és el conjunt dels mots acceptats.

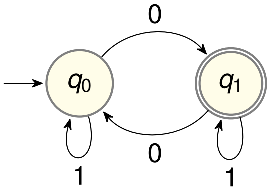
Qüestió

Quin és el llenguatge reconegut per l'autòmat finit anterior?

En un context formal, l'acció que permet fer una transició és la lectura d'un símbol d'un alfabet.

Un mot és acceptat si, partint de l'estat inicial, la lectura dels seus símbols permet arribar a un estat acceptador a través de transicions.

Exemple: autòmat finit sobre alfabet $\{0, 1\}$

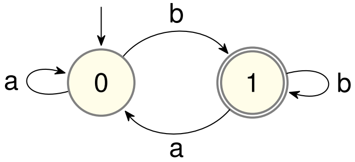


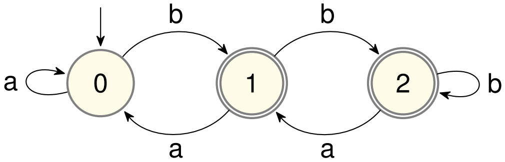
El llenguatge reconegut per un autòmat finit és el conjunt dels mots acceptats.

Qüestió

Quin és el llenguatge reconegut per l'autòmat finit anterior?

Considerem ara una versió més abstracta dels autòmats finits que simulen torniquets vistos abans.

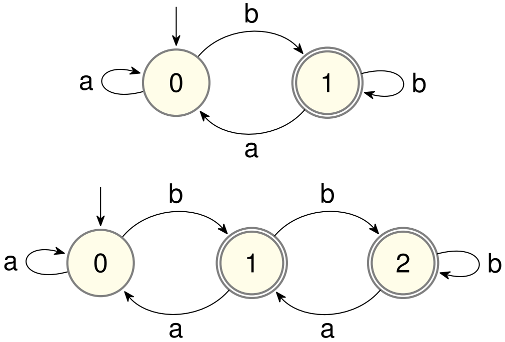




Qüestió

Quins són els llenguatges reconeguts pels autòmats anteriors?

Considerem ara una versió més abstracta dels autòmats finits que simulen torniquets vistos abans.



Qüestió

Quins són els llenguatges reconeguts pels autòmats anteriors?

Exercicis

Descriviu autòmats finits per als llenguatges següents sobre alfabet $\{a, b\}$:

1. $\emptyset$
2. $\{\lambda\}$
3. $\{a, b\}^*$
4. $\{a, b\}^+$
5. $\{a, b\}^{\leq 2}$
6. $\{a, b\}^3$

Exercicis

Descriviu informalment els llenguatges següents i trobeu autòmats finits que els reconeguin (en blau, el problema corresponent a la llista DFA (RACSO)):

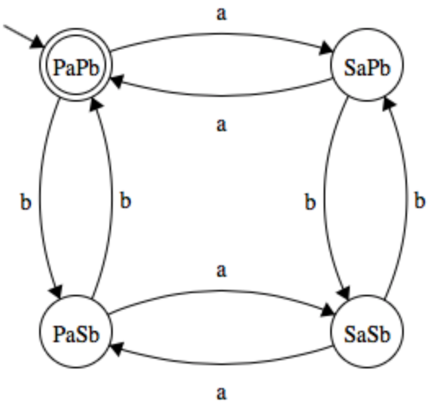
1. (2) $\{w \in \{a,b\}^* \mid |w|_a \in \dot{2} \ \wedge \ |w|_b \in \dot{2}\}$
2. (4) $\{w \in \{a,b\}^* \mid \exists x \;\; w = xa\}$
3. (8) $\{w \in \{a,b\}^* \mid \forall x,y \;\; w = xay \Rightarrow |x|_b \in \dot{2}\}$
4. (15) $\{w \in \{a,b\}^* \mid |w|_{bbb} = 0\}$
5. (20) $\{w \in \{0,1\}^* \mid \textbf{value}_2(w) \in \dot{2}\}$
6. (21) $\{w \in \{0,1\}^* \mid \textbf{value}_2(w) \in \dot{3}\}$

Notes

- $\textbf{value}_2(w)$ és el valor del nombre binari representat pel mot w
- Definim $\dot{n} = \{\ k \in \mathbb{N} \mid \exists i \in \mathbb{N} \;\; n \cdot i = k\ \}$

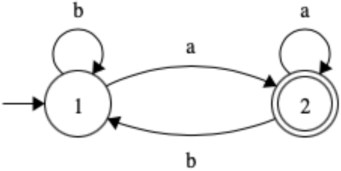
1. Exercici 2

$$\{w \in \{a, b\}^* \mid |w|_a \in \dot{2} \ \wedge \ |w|_b \in \dot{2}\}$$



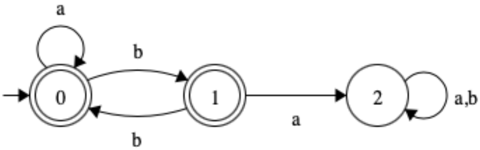
2. Exercici 4

$$\{w \in \{a, b\}^* \mid \exists x \ \ w = xa\}$$



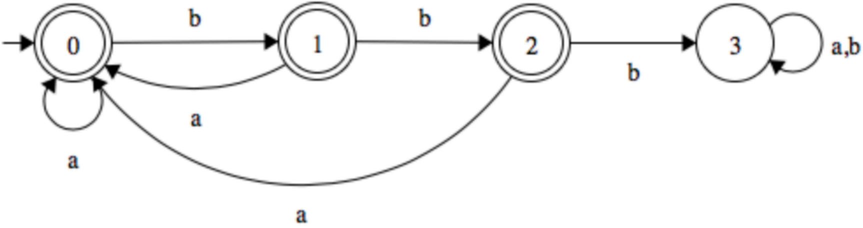
3. Exercici 8

$$\{w \in \{a, b\}^* \mid \forall x, y \;\; w = xay \Rightarrow |x|_b \in \dot{2}\}$$



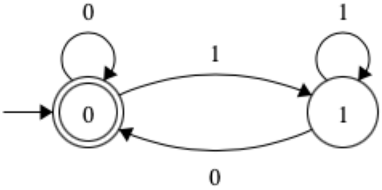
4. Exercici 15

$$\{w \in \{a, b\}^* \mid |w|_{bbb} = 0\}$$



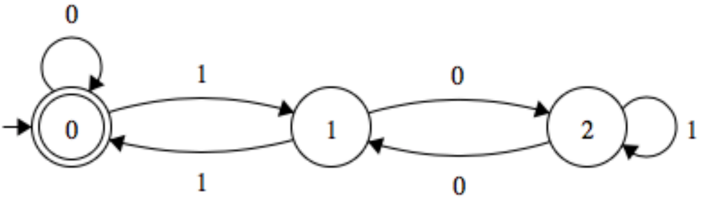
5. Exercici 20

$$\{w \in \{0,1\}^* \mid \textbf{value}_2(w) \in \dot{2}\}$$



6. Exercici 21

$$\{w \in \{0,1\}^* \mid \textbf{value}_2(w) \in \dot{3}\}$$



Autòmats finits

1 Concepte

2 Exemples

3 Definició formal

4 Llenguatge reconegut

5 Tancament per operacions 1

6 Indeterminisme

7 Tancament per operacions 2

Els autòmats finits són fàcils d'entendre amb el diagrama d'estats. Tot i així, hi ha motius per introduir la definició formal:

1. Guanyar precisió: qualsevol dubte s'ha de poder resoldre amb la definició.
2. Guanyar expressivitat: la notació serveix per expressar idees complexes clarament, especialment en les demostracions.

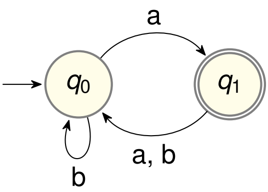
Definició

Un *autòmat finit determinista* (DFA, de *deterministic finite automaton*) és un quíntuple $(Q, \Sigma, \delta, q_0, F)$, on:

- Q és un conjunt finit d'elements anomenats estats
- Σ és un alfabet,
- $\delta : Q \times \Sigma \to Q$ és una funció total, anomenada funció de transició,
- $q_0 \in Q$ és l'estat inicial i
- $F \subseteq Q$ és el conjunt d'estats acceptadors.

Exemple: definició formal de DFA

Considereu l'autòmat finit S_a representat pel diagrama d'estats següent:



Podem descriure formalment S_a com:

$$S_a = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_1\}),$$

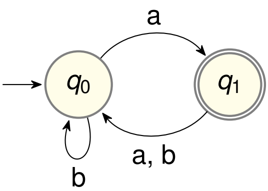
on la funció de transició δ es pot representar amb la taula

	a	b
q_0	q_1	q_0
q_1	q_0	q_0

És a dir, $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_0$, $\delta(q_1, a) = q_0$ i $\delta(q_1, b) = q_0$.

Exemple: definició formal de DFA

Considereu l'autòmat finit S_a representat pel diagrama d'estats següent:



Podem descriure formalment S_a com:

$$S_a = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_1\}),$$

on la funció de transició δ es pot representar amb la taula

	a	b
q_0	q_1	q_0
q_1	q_0	q_0

És a dir, $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_0$, $\delta(q_1, a) = q_0$ i $\delta(q_1, b) = q_0$.

Qüestions (respongueu tenint en compte la definició de sota)

1. És possible que un DFA no tingui estats acceptadors?
2. És possible que un estat no tingui cap transició a un altre? O que en tingui més d'una?

Definició (recordatori)

Un *autòmat finit determinista* (DFA, de *deterministic finite automaton*) és un quíntuple $(Q, \Sigma, \delta, q_0, F)$, on:

- Q és un conjunt finit d'elements anomenats estats
- Σ és un alfabet,
- $\delta : Q \times \Sigma \to Q$ és una funció total, anomenada funció de transició,
- $q_0 \in Q$ és l'estat inicial i
- $F \subseteq Q$ és el conjunt d'estats acceptadors.

Autòmats finits

1 Concepte

2 Exemples

3 Definició formal

4 Llenguatge reconegut

5 Tancament per operacions 1

6 Indeterminisme

7 Tancament per operacions 2

Funció de transició estesa

Sigui $M = (Q, \Sigma, \delta, q_0, F)$ un DFA. La funció de transició estesa de M és la funció $\tilde{\delta} : Q \times \Sigma^* \to Q$ tal que, per a $q \in Q$ i $x \in \Sigma^*$:

$$\tilde{\delta}(q, x) = \begin{cases} q, & \text{si } x = \lambda, \\ \tilde{\delta}(\delta(q, a), y), & \text{si } x = ay, \text{amb } a \in \Sigma. \end{cases}$$

Exemple

Podem calcular $\tilde{\delta}$ sobre *aba* en el DFA S_a vist abans

$$\begin{aligned} \tilde{\delta}(q_0, aba) &= \tilde{\delta}(\delta(q_0, a), ba) = \tilde{\delta}(q_1, ba) = \\ &= \tilde{\delta}(\delta(q_1, b), a) = \tilde{\delta}(q_0, a) = \\ &= \tilde{\delta}(\delta(q_0, a), \lambda) = \tilde{\delta}(q_1, \lambda) = q_1 \end{aligned}$$

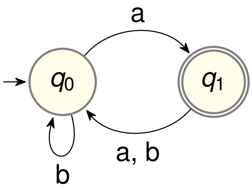
Funció de transició estesa

Sigui $M = (Q, \Sigma, \delta, q_0, F)$ un DFA. La funció de transició estesa de M és la funció $\tilde{\delta} : Q \times \Sigma^* \to Q$ tal que, per a $q \in Q$ i $x \in \Sigma^*$:

$$\tilde{\delta}(q, x) = \begin{cases} q, & \text{si } x = \lambda, \\ \tilde{\delta}(\delta(q, a), y), & \text{si } x = ay, \text{amb } a \in \Sigma. \end{cases}$$

Exemple

Podem calcular $\tilde{\delta}$ sobre *aba* en el DFA S_a vist abans



$$\begin{aligned} \tilde{\delta}(q_0, aba) &= \tilde{\delta}(\delta(q_0, a), ba) = \tilde{\delta}(q_1, ba) = \\ &= \tilde{\delta}(\delta(q_1, b), a) = \tilde{\delta}(q_0, a) = \\ &= \tilde{\delta}(\delta(q_0, a), \lambda) = \tilde{\delta}(q_1, \lambda) = q_1 \end{aligned}$$

Notació

Habitualment farem servir el mateix símbol de la funció de transició per representar la funció de transició estesa.

Definició de llenguatge reconegut

Sigui $M = (Q, \Sigma, \delta, q_0, F)$ un DFA. Diem que M accepta un mot $x \in \Sigma^*$ si $\delta(q_0, x) \in F$. El llenguatge reconegut per M es defineix com

$$L(M) = \{x \in \Sigma^* \mid M \text{ accepta } x\}.$$

És a dir, $L(M)$ és el llenguatge format pels mots sobre Σ que són acceptats per M.

Notació

Habitualment farem servir el mateix símbol de la funció de transició per representar la funció de transició estesa.

Definició de llenguatge reconegut

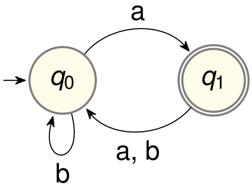
Sigui $M = (Q, \Sigma, \delta, q_0, F)$ un DFA. Diem que M accepta un mot $x \in \Sigma^*$ si $\delta(q_0, x) \in F$. El llenguatge reconegut per M es defineix com

$$L(M) = \{x \in \Sigma^* \mid M \text{ accepta } x\}.$$

És a dir, $L(M)$ és el llenguatge format pels mots sobre Σ que són acceptats per M.

Exemple: llenguatge reconegut

Recordem l'autòmat finit S_a:



Hem vist que $aba \in L(S_a)$. Els mots acceptats són:

- a, a^3, a^5, ...
- wba, wba^3, wba^5, ... per a qualsevol mot $w \in \{a, b\}^*$

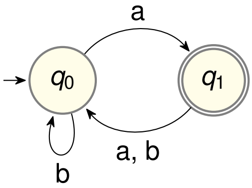
Per tant, $L(S_a)$ consisteix en tots els mots sobre $\{a, b\}$ que acaben en un nombre senar d'as.

Exercici

Doneu una definició formal de $L(S_a)$.

Exemple: llenguatge reconegut

Recordem l'autòmat finit S_a:



Hem vist que $aba \in L(S_a)$. Els mots acceptats són:

- $a, a^3, a^5, \ldots$
- $wba, wba^3, wba^5, \ldots$ per a qualsevol mot $w \in \{a, b\}^*$

Per tant, $L(S_a)$ consisteix en tots els mots sobre $\{a, b\}$ que acaben en un nombre senar d'as.

Exercici

Doneu una definició formal de $L(S_a)$.

Llenguatges regulars

Un llenguatge se'n diu regular si existeix un DFA que el reconeix.

És a dir, L és regular si existeix un DFA M tal que $L = L(M)$.

Exemple: nombre parell d'as

El llenguatge

$$\{w \in \{a, b\}^* \mid |w|_a \in \dot{2}\}$$

dels mots sobre $\{a, b\}$ que tenen un nombre parell d'as és regular perquè el DFA següent el reconeix:

Llenguatges regulars

Un llenguatge se'n diu regular si existeix un DFA que el reconeix.

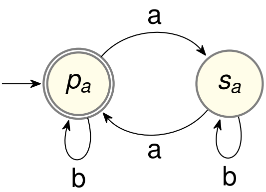
És a dir, L és regular si existeix un DFA M tal que $L = L(M)$.

Exemple: nombre parell d'as

El llenguatge

$$\{w \in \{a,b\}^* \mid |w|_a \in \dot{2}\}$$

dels mots sobre $\{a,b\}$ que tenen un nombre parell d'as és regular perquè el DFA següent el reconeix:



Autòmats finits

1. Concepte
2. Exemples
3. Definició formal
4. Llenguatge reconegut
5. Tancament per operacions 1
6. Indeterminisme
7. Tancament per operacions 2

Sigui $\mathbb{N} = \{1, 2, 3, \dots\}$ el conjunt dels nombres naturals.

Diem que el conjunt $\mathbb{N}$ és tancat per la suma perquè per qualsevol parell de naturals i, j, el nombre $i + j$ també és a $\mathbb{N}$.

Qüestió

Per quines operacions aritmètiques és tancat $\mathbb{N}$?

Sigui $\mathbb{N} = \{1, 2, 3, \dots\}$ el conjunt dels nombres naturals.

Diem que el conjunt $\mathbb{N}$ és tancat per la suma perquè per qualsevol parell de naturals i, j, el nombre $i + j$ també és a $\mathbb{N}$.

Qüestió

Per quines operacions aritmètiques és tancat $\mathbb{N}$?

En general, donada una col·lecció d'objectes $\mathcal{C}$, diem que $\mathcal{C}$ és tancada per una operació $\circ$ si aplicar $\circ$ a objectes de $\mathcal{C}$ ens retorna un objecte de $\mathcal{C}$.

Conjunts de parts

Donat un conjunt A, es defineix

1. $\mathcal{P}(A) = \{S \subseteq A\}$
2. $\mathcal{P}_F(A) = \{S \subseteq A \mid S \text{ és finit}\}$
3. $\mathcal{P}_I(A) = \{S \subseteq A \mid S \text{ és infinit}\}$

Qüestió

Per quines operacions (de reunió, intersecció i complementació) són tancades les famílies $\mathcal{P}(\mathbb{N})$, $\mathcal{P}_F(\mathbb{N})$ i $\mathcal{P}_I(\mathbb{N})$?

En general, donada una col·lecció d'objectes $\mathcal{C}$, diem que $\mathcal{C}$ és tancada per una operació $\circ$ si aplicar $\circ$ a objectes de $\mathcal{C}$ ens retorna un objecte de $\mathcal{C}$.

Conjunts de parts

Donat un conjunt A, es defineix

1. $\mathcal{P}(A) = \{S \subseteq A\}$
2. $\mathcal{P}_F(A) = \{S \subseteq A \mid S \text{ és finit } \}$
3. $\mathcal{P}_I(A) = \{S \subseteq A \mid S \text{ és infinit } \}$

Qüestió

Per quines operacions (de reunió, intersecció i complementació) són tancades les famílies $\mathcal{P}(\mathbb{N})$, $\mathcal{P}_F(\mathbb{N})$ i $\mathcal{P}_I(\mathbb{N})$?

En general, donada una col·lecció d'objectes $\mathcal{C}$, diem que $\mathcal{C}$ és tancada per una operació $\circ$ si aplicar $\circ$ a objectes de $\mathcal{C}$ ens retorna un objecte de $\mathcal{C}$.

Conjunts de parts

Donat un conjunt A, es defineix

1. $\mathcal{P}(A) = \{S \subseteq A\}$
2. $\mathcal{P}_F(A) = \{S \subseteq A \mid S \text{ és finit }\}$
3. $\mathcal{P}_I(A) = \{S \subseteq A \mid S \text{ és infinit }\}$

Qüestió

Per quines operacions (de reunió, intersecció i complementació) són tancades les famílies $\mathcal{P}(\mathbb{N})$, $\mathcal{P}_F(\mathbb{N})$ i $\mathcal{P}_I(\mathbb{N})$?

La nostra col·lecció serà la següent.

Llenguatges regulars

Anomenem $\textsc{Reg}$ la classe dels llenguatges regulars.

Observació

La família de tots els llenguatges sobre un alfabet Σ és $\mathcal{P}(\Sigma^*)$. Així doncs, $\textsc{Reg} \subseteq \mathcal{P}(\Sigma^*)$.

Veurem que la classe $\textsc{Reg}$ és tancada per les operacions de complementació, intersecció, reunió, concatenació, estrella de Kleene i tancament positiu.

La nostra col·lecció serà la següent.

Llenguatges regulars

Anomenem REG la classe dels llenguatges regulars.

Observació

La família de tots els llenguatges sobre un alfabet Σ és $\mathcal{P}(\Sigma^*)$. Així doncs, REG $\subseteq \mathcal{P}(\Sigma^*)$.

Veurem que la classe REG és tancada per les operacions de complementació, intersecció, reunió, concatenació, estrella de Kleene i tancament positiu.

La nostra col·lecció serà la següent.

Llenguatges regulars

Anomenem $\textsc{Reg}$ la classe dels llenguatges regulars.

Observació

La família de tots els llenguatges sobre un alfabet Σ és $\mathcal{P}(\Sigma^*)$. Així doncs, $\textsc{Reg} \subseteq \mathcal{P}(\Sigma^*)$.

Veurem que la classe $\textsc{Reg}$ és tancada per les operacions de complementació, intersecció, reunió, concatenació, estrella de Kleene i tancament positiu.

Complementació

REG és tancada per complementació

Suposem que $L \in$REG. Per tant, existeix un DFA $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L = L(A)$.

Per reconèixer $\overline{L}$, definim el DFA que té els estats acceptadors intercanviats amb els no acceptadors:

$$B = (Q, \Sigma, \delta, q_0, Q \setminus F).$$

Tenim que per a tot $x \in \Sigma^*$,

$$x \in L(B) \Leftrightarrow \delta(q_0, x) \in Q \setminus F \Leftrightarrow \delta(q_0, x) \notin F \Leftrightarrow x \notin L(A) \Leftrightarrow x \in \overline{L(A)}$$

i, per tant, $L(B) = \overline{L(A)} = \overline{L}$, és a dir, el DFA B reconeix $\overline{L}$.
Aleshores, $\overline{L} \in$REG.

Complementació

REG és tancada per complementació

Suposem que $L \in$REG. Per tant, existeix un DFA $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L = L(A)$.

Per reconèixer $\overline{L}$, definim el DFA que té els estats acceptadors intercanviats amb els no acceptadors:

$$B = (Q, \Sigma, \delta, q_0, Q \setminus F).$$

Tenim que per a tot $x \in \Sigma^*$,

$$x \in L(B) \Leftrightarrow \delta(q_0, x) \in Q \setminus F \Leftrightarrow \delta(q_0, x) \notin F \Leftrightarrow x \notin L(A) \Leftrightarrow x \in \overline{L(A)}$$

i, per tant, $L(B) = \overline{L(A)} = \overline{L}$, és a dir, el DFA B reconeix $\overline{L}$.
Aleshores, $\overline{L} \in$REG.

Complementació

REG és tancada per complementació

Suposem que $L \in$REG. Per tant, existeix un DFA $A = (Q, \Sigma, \delta, q_0, F)$ tal que $L = L(A)$.

Per reconèixer $\overline{L}$, definim el DFA que té els estats acceptadors intercanviats amb els no acceptadors:

$$B = (Q, \Sigma, \delta, q_0, Q \setminus F).$$

Tenim que per a tot $x \in \Sigma^*$,

$$x \in L(B) \Leftrightarrow \delta(q_0, x) \in Q \setminus F \Leftrightarrow \delta(q_0, x) \notin F \Leftrightarrow x \notin L(A) \Leftrightarrow x \in \overline{L(A)}$$

i, per tant, $L(B) = \overline{L(A)} = \overline{L}$, és a dir, el DFA B reconeix $\overline{L}$.
Aleshores, $\overline{L} \in$REG.

Complementació

Exemple: tancament per complementació

Recordem l'exercici 8 de RACSO per al qual es demanava un DFA que el reconegués: $L = \{w \in \{a,b\}^* \mid \forall x,y \;\; w = xay \Rightarrow |x|_b \in \dot{2}\}$.

Fixem-nos que conceptualment pot ser més senzill trobar un DFA per a

$$\overline{L} = \{w \in \{a,b\}^* \mid \exists x,y \;\; w = xay \wedge |x|_b \notin \dot{2}\}.$$

El motiu és que només cal assegurar que s'accepten els mots per als quals existeix una descomposició com la descrita.

El DFA que reconeix L s'obté intercanviant estats acceptadors i no acceptadors com es descriu al teorema.

Complementació

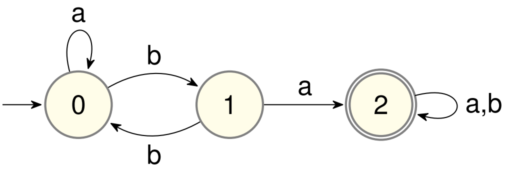
Exemple: tancament per complementació

Recordem l'exercici 8 de RACSO per al qual es demanava un DFA que el reconegués: $L = \{w \in \{a,b\}^* \mid \forall x,y \;\; w = xay \Rightarrow |x|_b \in \dot{2}\}$.

Fixem-nos que conceptualment pot ser més senzill trobar un DFA per a

$$\overline{L} = \{w \in \{a,b\}^* \mid \exists x,y \;\; w = xay \wedge |x|_b \notin \dot{2}\}.$$

El motiu és que només cal assegurar que s'accepten els mots per als quals existeix una descomposició com la descrita.



El DFA que reconeix L s'obté intercanviant estats acceptadors i no acceptadors com es descriu al teorema.

Intersecció

Suposem que, donats dos llenguatges regulars L_1 i L_2, volem demostrar que $L_1 \cap L_2$ és regular. Volem fer-ho de forma constructiva, combinant dos DFAs que reconeguin L_1 i L_2, diguem M_1 i M_2, resp.

Com podríem simular M_1 i M_2 per reconèixer $L_1 \cap L_2$?

1. De forma consecutiva, per exemple M_1 seguit de M_2?
   Un cop llegit un símbol amb M_1, ja no està disponible per a M_2!

2. De forma simultània. Recordem dos estats en tot moment, inicialment els dos estats inicials. Llavors,

   - llegim un símbol d'entrada
   - calculem la transició de cada estat del parell amb aquest símbol

   Quan s'han acabat els símbols, acceptem si tant M_1 com M_2 han acceptat l'entrada.

Intersecció

Suposem que, donats dos llenguatges regulars L_1 i L_2, volem demostrar que $L_1 \cap L_2$ és regular. Volem fer-ho de forma constructiva, combinant dos DFAs que reconeguin L_1 i L_2, diguem M_1 i M_2, resp.

Com podríem simular M_1 i M_2 per reconèixer $L_1 \cap L_2$?

1. De forma consecutiva, per exemple M_1 seguit de M_2?
   Un cop llegit un símbol amb M_1, ja no està disponible per a M_2!
2. De forma simultània. Recordem dos estats en tot moment, inicialment els dos estats inicials. Llavors,
   - llegim un símbol d'entrada
   - calculem la transició de cada estat del parell amb aquest símbol

   Quan s'han acabat els símbols, acceptem si tant M_1 com M_2 han acceptat l'entrada.

Intersecció

Suposem que, donats dos llenguatges regulars L_1 i L_2, volem demostrar que $L_1 \cap L_2$ és regular. Volem fer-ho de forma constructiva, combinant dos DFAs que reconeguin L_1 i L_2, diguem M_1 i M_2, resp.

Com podríem simular M_1 i M_2 per reconèixer $L_1 \cap L_2$?

1. De forma consecutiva, per exemple M_1 seguit de M_2?
   Un cop llegit un símbol amb M_1, ja no està disponible per a M_2!
2. De forma simultània. Recordem dos estats en tot moment, inicialment els dos estats inicials. Llavors,
   - llegim un símbol d'entrada
   - calculem la transició de cada estat del parell amb aquest símbol

   Quan s'han acabat els símbols, acceptem si tant M_1 com M_2 han acceptat l'entrada.

Intersecció

Reg és tancada per intersecció

Suposem que $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ i $M_2 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ són dos DFAs tals que $L(M_1) = L_1$ i $L(M_2) = L_2$. Definim un DFA que pot simular M_1 i M_2 alhora:

$$M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2),$$

on $\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$ per a $p \in Q_1$, $q \in Q_2$ i $a \in \Sigma$.

És fàcil veure per inducció que $\forall w \in \Sigma^* \; \forall p \in Q_1 \; \forall q \in Q_2$,

$$\delta((p, q), w) = (\delta_1(p, w), \delta_2(q, w)).$$

Llavors, donat un mot $w \in \Sigma^*$,

$$\begin{aligned} w \in L(M) &\Leftrightarrow \delta((q_1, q_2), w) \in F_1 \times F_2 \\ &\Leftrightarrow (\delta_1(q_1, w), \delta_2(q_2, w)) \in F_1 \times F_2 \\ &\Leftrightarrow w \in L_1 \;\wedge\; w \in L_2 \\ &\Leftrightarrow w \in L_1 \cap L_2. \end{aligned}$$

Per tant, $L_1 \cap L_2$ és reconegut pel DFA M i, aleshores, és regular.

Intersecció

REG és tancada per intersecció

Suposem que $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ i $M_2 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ són dos DFAs tals que $L(M_1) = L_1$ i $L(M_2) = L_2$. Definim un DFA que pot simular M_1 i M_2 alhora:

$$M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2),$$

on $\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$ per a $p \in Q_1$, $q \in Q_2$ i $a \in \Sigma$.

És fàcil veure per inducció que $\forall w \in \Sigma^* \ \forall p \in Q_1 \ \forall q \in Q_2$,

$$\delta((p, q), w) = (\delta_1(p, w), \delta_2(q, w)).$$

Llavors, donat un mot $w \in \Sigma^*$,

$$\begin{aligned} w \in L(M) &\Leftrightarrow \delta((q_1, q_2), w) \in F_1 \times F_2 \\ &\Leftrightarrow (\delta_1(q_1, w), \delta_2(q_2, w)) \in F_1 \times F_2 \\ &\Leftrightarrow w \in L_1 \ \wedge \ w \in L_2 \\ &\Leftrightarrow w \in L_1 \cap L_2. \end{aligned}$$

Per tant, $L_1 \cap L_2$ és reconegut pel DFA M i, aleshores, és regular.

Intersecció

REG és tancada per intersecció

Suposem que $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ i $M_2 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ són dos DFAs tals que $L(M_1) = L_1$ i $L(M_2) = L_2$. Definim un DFA que pot simular M_1 i M_2 alhora:

$$M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2),$$

on $\delta((p,q),a) = (\delta_1(p,a), \delta_2(q,a))$ per a $p \in Q_1$, $q \in Q_2$ i $a \in \Sigma$.

És fàcil veure per inducció que $\forall w \in \Sigma^* \; \forall p \in Q_1 \; \forall q \in Q_2$,

$$\delta((p,q),w) = (\delta_1(p,w), \delta_2(q,w)).$$

Llavors, donat un mot $w \in \Sigma^*$,

$$\begin{aligned} w \in L(M) &\Leftrightarrow \delta((q_1,q_2),w) \in F_1 \times F_2 \\ &\Leftrightarrow (\delta_1(q_1,w), \delta_2(q_2,w)) \in F_1 \times F_2 \\ &\Leftrightarrow w \in L_1 \;\wedge\; w \in L_2 \\ &\Leftrightarrow w \in L_1 \cap L_2. \end{aligned}$$

Per tant, $L_1 \cap L_2$ és reconegut pel DFA M i, aleshores, és regular.

Intersecció

REG és tancada per intersecció

Suposem que $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ i $M_2 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ són dos DFAs tals que $L(M_1) = L_1$ i $L(M_2) = L_2$. Definim un DFA que pot simular M_1 i M_2 alhora:

$$M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2),$$

on $\delta((p,q),a) = (\delta_1(p,a), \delta_2(q,a))$ per a $p \in Q_1$, $q \in Q_2$ i $a \in \Sigma$.

És fàcil veure per inducció que $\forall w \in \Sigma^* \; \forall p \in Q_1 \; \forall q \in Q_2$,

$$\delta((p,q),w) = (\delta_1(p,w), \delta_2(q,w)).$$

Llavors, donat un mot $w \in \Sigma^*$,

$$\begin{aligned} w \in L(M) &\Leftrightarrow \delta((q_1,q_2),w) \in F_1 \times F_2 \\ &\Leftrightarrow (\delta_1(q_1,w), \delta_2(q_2,w)) \in F_1 \times F_2 \\ &\Leftrightarrow w \in L_1 \;\wedge\; w \in L_2 \\ &\Leftrightarrow w \in L_1 \cap L_2. \end{aligned}$$

Per tant, $L_1 \cap L_2$ és reconegut pel DFA M i, aleshores, és regular.

Intersecció

Exemple: tancament per intersecció

Recordem ara l'exercici 2 del RACSO on es demanava un DFA per al llenguatge

$$L = \{w \in \{a,b\}^* \mid |w|_a \in \dot{2} \ \wedge \ |w|_b \in \dot{2}\}.$$

Donat un símbol $s \in \{a,b\}$, definim $L_s = \{w \in \{a,b\}^* \mid |w|_s \in \dot{2}\}$.

Intersecció

Exemple: tancament per intersecció

Recordem ara l'exercici 2 del RACSO on es demanava un DFA per al llenguatge

$$L = \{w \in \{a, b\}^* \mid |w|_a \in \dot{2} \ \wedge \ |w|_b \in \dot{2}\}.$$

Donat un símbol $s \in \{a, b\}$, definim $L_s = \{w \in \{a, b\}^* \mid |w|_s \in \dot{2}\}$.

Intersecció

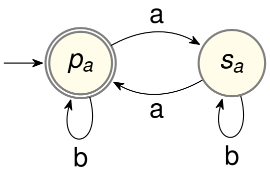
Exemple: tancament per intersecció

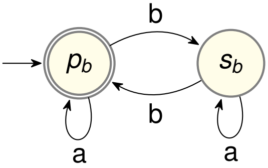
Recordem ara l'exercici 2 del RACSO on es demanava un DFA per al llenguatge

$$L = \{w \in \{a, b\}^* \mid |w|_a \in \dot{2} \ \wedge \ |w|_b \in \dot{2}\}.$$

Donat un símbol $s \in \{a, b\}$, definim $L_s = \{w \in \{a, b\}^* \mid |w|_s \in \dot{2}\}$.

Llavors, clarament $L = L_a \cap L_b$, on L_a i L_b són reconeguts pels DFAs





Intersecció

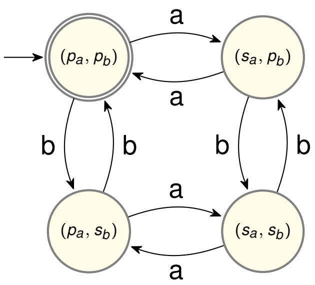
Exemple: tancament per intersecció

Recordem ara l'exercici 2 del RACSO on es demanava un DFA per al llenguatge

$$L = \{w \in \{a, b\}^* \mid |w|_a \in \dot{2} \ \wedge \ |w|_b \in \dot{2}\}.$$

Donat un símbol $s \in \{a, b\}$, definim $L_s = \{w \in \{a, b\}^* \mid |w|_s \in \dot{2}\}$.

El DFA intersecció és



Reunió

Reg és tancada per reunió

Siguin $L_1, L_2 \in \textsc{Reg}$. Llavors,

$$L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$$

és regular perquè les operacions de complementació i intersecció de la dreta de l'equació preserven la regularitat.

Exercici

Definiu formalment el conjunt d'estats acceptadors del DFA obtingut per producte cartesià per reconèixer la reunió dels llenguatges de dos DFAs donats.

Reunió

REG és tancada per reunió

Siguin $L_1, L_2 \in \text{REG}$. Llavors,

$$L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$$

és regular perquè les operacions de complementació i intersecció de la dreta de l'equació preserven la regularitat.

Exercici

Definiu formalment el conjunt d'estats acceptadors del DFA obtingut per producte cartesià per reconèixer la reunió dels llenguatges de dos DFAs donats.

Reunió

REG és tancada per reunió

Siguin $L_1, L_2 \in \textsc{Reg}$. Llavors,

$$L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$$

és regular perquè les operacions de complementació i intersecció de la dreta de l'equació preserven la regularitat.

Exercici

Definiu formalment el conjunt d'estats acceptadors del DFA obtingut per producte cartesià per reconèixer la reunió dels llenguatges de dos DFAs donats.

Altres operacions

Veurem que els regulars també són tancats per

- concatenació
- estrella de Kleene
- tancament positiu

Considerem la concatenació. Per demostrar que $L_1, L_2 \in \text{REG}$ implica $L_1 \cdot L_2 \in \text{REG}$, cal trobar un DFA que, donat un mot $w \in \Sigma^*$, decideixi si existeix una factorització de w de la forma

$$w = xy, \text{ on } x \in L_1 \text{ i } y \in L_2.$$

El problema és com saber quina és la factorització correcta! (també en el cas de l'estrella i del tancament positiu).

Per resoldre'l, introduïm una nova tècnica anomenada indeterminisme.

Altres operacions

Veurem que els regulars també són tancats per

- concatenació
- estrella de Kleene
- tancament positiu

Considerem la concatenació. Per demostrar que $L_1, L_2 \in \textsc{Reg}$ implica $L_1 \cdot L_2 \in \textsc{Reg}$, cal trobar un DFA que, donat un mot $w \in \Sigma^*$, decideixi si existeix una factorització de w de la forma

$$w = xy, \text{ on } x \in L_1 \text{ i } y \in L_2.$$

El problema és com saber quina és la factorització correcta! (també en el cas de l'estrella i del tancament positiu).

Per resoldre'l, introduïm una nova tècnica anomenada indeterminisme.

Altres operacions

Veurem que els regulars també són tancats per

- concatenació
- estrella de Kleene
- tancament positiu

Considerem la concatenació. Per demostrar que $L_1, L_2 \in \textsc{Reg}$ implica $L_1 \cdot L_2 \in \textsc{Reg}$, cal trobar un DFA que, donat un mot $w \in \Sigma^*$, decideixi si existeix una factorització de w de la forma

$$w = xy, \text{ on } x \in L_1 \text{ i } y \in L_2.$$

El problema és com saber quina és la factorització correcta! (també en el cas de l'estrella i del tancament positiu).

Per resoldre'l, introduïm una nova tècnica anomenada indeterminisme.

Autòmats finits

1 Concepte
2 Exemples
3 Definició formal
4 Llenguatge reconegut
5 Tancament per operacions 1
6 Indeterminisme
7 Tancament per operacions 2

Model indeterminista

L'indeterminisme és un dels conceptes que han tingut més repercussió en la teoria de la computació.

Hem anomenat determinista al model d'autòmat finit (**D**FA) perquè no hi ha cap marge de variació per la computació d'un autòmat amb una entrada donada:

- l'estat inicial és únic
- no està permès fer una transició sense llegir un símbol
- l'estat següent està determinat per l'estat i símbol actuals

Model indeterminista

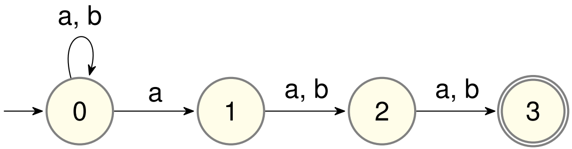
L'indeterminisme és un dels conceptes que han tingut més repercussió en la teoria de la computació.

Hem anomenat determinista al model d'autòmat finit (**D**FA) perquè no hi ha cap marge de variació per la computació d'un autòmat amb una entrada donada:

- l'estat inicial és únic
- no està permès fer una transició sense llegir un símbol
- l'estat següent està determinat per l'estat i símbol actuals

Model indeterminista

En el model indeterminista pot existir més d'una tria en una o més transicions.



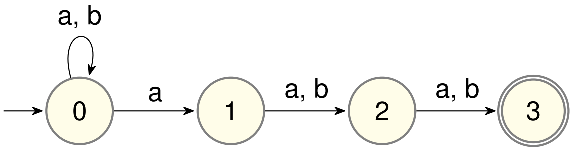
Un mot és acceptat quan existeix un camí de l'estat inicial a un estat acceptador.

Qüestió

Quin és el llenguatge reconegut per l'autòmat indeterminista anterior?

Model indeterminista

En el model indeterminista pot existir més d'una tria en una o més transicions.



Un mot és acceptat quan existeix un camí de l'estat inicial a un estat acceptador.

Qüestió

Quin és el llenguatge reconegut per l'autòmat indeterminista anterior?

Definicions

Definició

Un *autòmat finit indeterminista* (NFA, de *nondeterministic finite automaton*) és un quíntuple $(Q, \Sigma, \delta, I, F)$, on:

- Q és un conjunt finit d'estats
- Σ és un alfabet,
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ és una funció total (de transició),
- $I \subseteq Q$ és el conjunt d'estats inicials i
- $F \subseteq Q$ és el conjunt d'estats acceptadors.

Exercici

Descriviu formalment l'NFA de l'exemple anterior.

Definicions

Definició

Un *autòmat finit indeterminista* (NFA, de *nondeterministic finite automaton*) és un quíntuple $(Q, \Sigma, \delta, I, F)$, on:

- Q és un conjunt finit d'estats
- Σ és un alfabet,
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ és una funció total (de transició),
- $I \subseteq Q$ és el conjunt d'estats inicials i
- $F \subseteq Q$ és el conjunt d'estats acceptadors.

Exercici

Descriviu formalment l'NFA de l'exemple anterior.

Definició de funció de transició estesa

Sigui $M = (Q, \Sigma, \delta, I, F)$ un NFA. La funció de transició estesa de M és la funció $\tilde{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$ tal que, per a $q \in Q$ i $x \in \Sigma^*$:

$$\tilde{\delta}(q, x) = \begin{cases} q, & \text{si } x = \lambda, \\ \bigcup_{p \in \delta(q,a)} \tilde{\delta}(p, y), & \text{si } x = ay, \text{amb } a \in \Sigma. \end{cases}$$

Notació

Com en el cas determinista, es fa servir habitualment δ en lloc de $\tilde{\delta}$.

Exemple

Calculeu $\delta(0, aabb)$ en l'NFA següent.

Definició de funció de transició estesa

Sigui $M = (Q, \Sigma, \delta, I, F)$ un NFA. La funció de transició estesa de M és la funció $\tilde{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$ tal que, per a $q \in Q$ i $x \in \Sigma^*$:

$$\tilde{\delta}(q, x) = \begin{cases} q, & \text{si } x = \lambda, \\ \bigcup_{p \in \delta(q,a)} \tilde{\delta}(p, y), & \text{si } x = ay, \text{amb } a \in \Sigma. \end{cases}$$

Notació

Com en el cas determinista, es fa servir habitualment δ en lloc de $\tilde{\delta}$.

Exemple

Calculeu $\delta(0, aabb)$ en l'NFA següent.

Definició de funció de transició estesa

Sigui $M = (Q, \Sigma, \delta, I, F)$ un NFA. La funció de transició estesa de M és la funció $\tilde{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$ tal que, per a $q \in Q$ i $x \in \Sigma^*$:

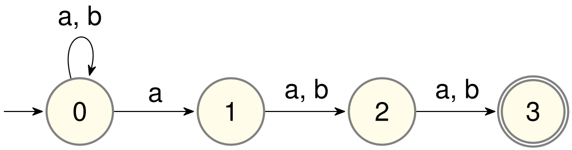
$$\tilde{\delta}(q, x) = \begin{cases} q, & \text{si } x = \lambda, \\ \bigcup_{p \in \delta(q,a)} \tilde{\delta}(p, y), & \text{si } x = ay, \text{amb } a \in \Sigma. \end{cases}$$

Notació

Com en el cas determinista, es fa servir habitualment δ en lloc de $\tilde{\delta}$.

Exemple

Calculeu $\delta(0, aabb)$ en l'NFA següent.



Definició de llenguatge reconegut

Sigui $M = (Q, \Sigma, \delta, q_0, F)$ un NFA. Diem que M accepta un mot $x \in \Sigma^*$ si $\delta(q_0, x) \cap F \neq \emptyset$ per a algun estat $q_0 \in I$. El llenguatge reconegut per M es defineix com

$$L(M) = \{x \in \Sigma^* \mid M \text{ accepta } x\}.$$

És a dir, $L(M)$ és el llenguatge format pels mots sobre Σ que són acceptats per M.

Qüestió

Quin és el llenguatge reconegut per l'NFA següent?

Definició de llenguatge reconegut

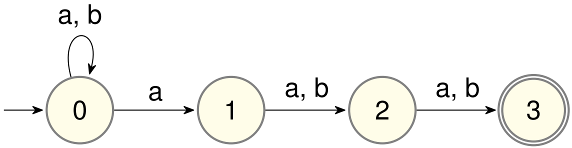
Sigui $M = (Q, \Sigma, \delta, q_0, F)$ un NFA. Diem que M accepta un mot $x \in \Sigma^*$ si $\delta(q_0, x) \cap F \neq \emptyset$ per a algun estat $q_0 \in I$. El llenguatge reconegut per M es defineix com

$$L(M) = \{x \in \Sigma^* \mid M \text{ accepta } x\}.$$

És a dir, $L(M)$ és el llenguatge format pels mots sobre Σ que són acceptats per M.

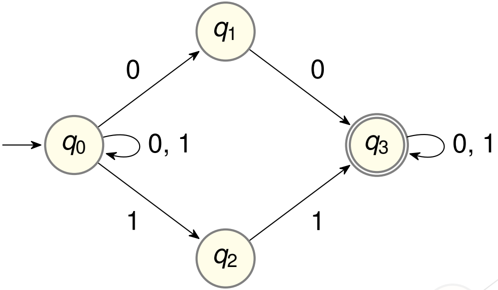
Qüestió

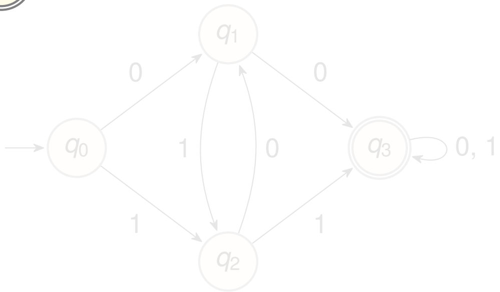
Quin és el llenguatge reconegut per l'NFA següent?



Equivalència entre NFAs i DFAs

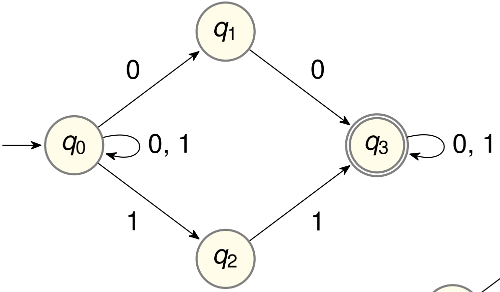
Tot NFA té un DFA equivalent.



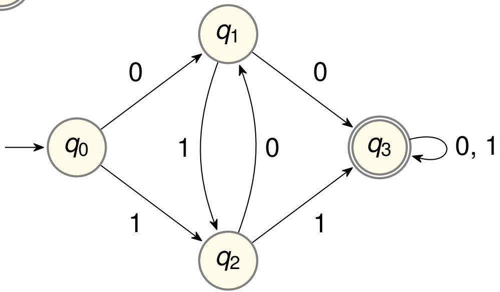


Equivalència entre NFAs i DFAs

Tot NFA

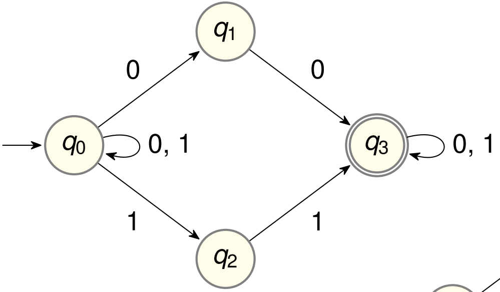


té un DFA equivalent.

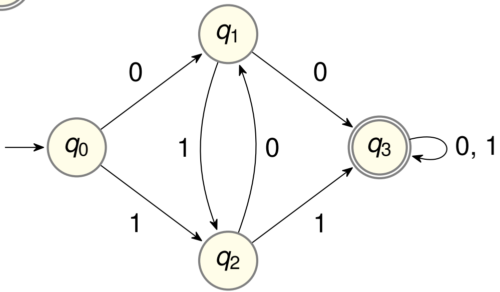


Equivalència entre NFAs i DFAs

Tot NFA



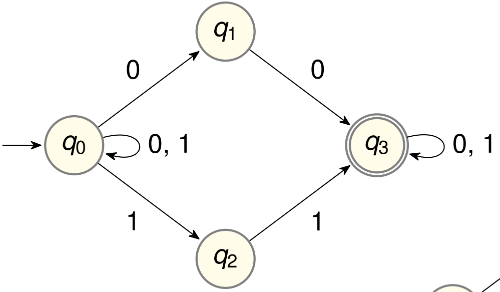
té un DFA equivalent.



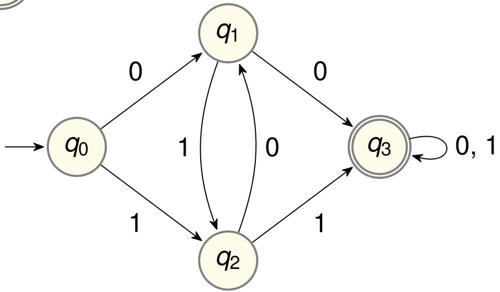
Però la transformació pot fer créixer el nombre d'estats exponencialment.

Equivalència entre NFAs i DFAs

Tot NFA

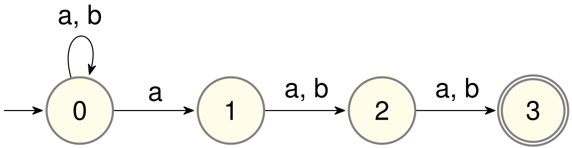


té un DFA equivalent.



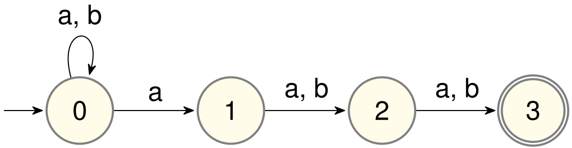
Qüestió: quin és el llenguatge reconegut per aquests autòmats?

Exemple: NFA

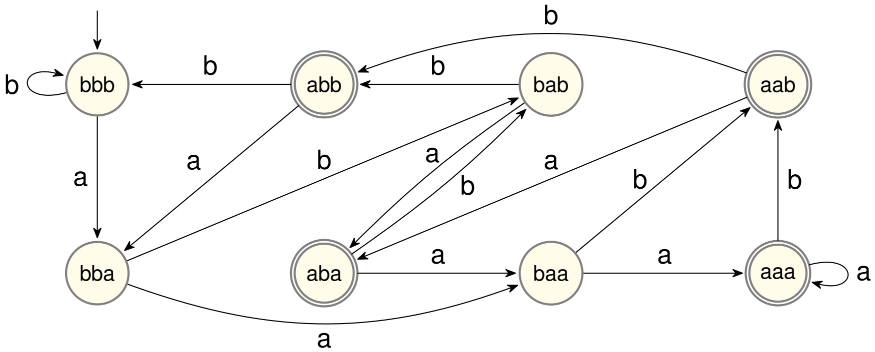


DFA equivalent

Exemple: NFA



DFA equivalent



Teorema

Tot NFA té un DFA equivalent.

Donat un NFA $M = (Q, \Sigma, \delta, I, F)$, definirem un DFA que tindrà com a estats tots els subconjunts d'estats de M, és a dir, $\mathcal{P}(Q)$. Definim el DFA

$$M' = (\mathcal{P}(Q), \Sigma, \delta', I, F'),$$

on per a tot $S \subseteq Q$, $a \in \Sigma$

$$\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$$

i $S \in F'$ si i només si S conté algun estat de F, és a dir,

$$F' = \{S \in \mathcal{P}(Q) \mid S \cap F \neq \emptyset\}.$$

Així, la transformació és $M = (Q, \Sigma, \delta, I, F) \longrightarrow M' = (\mathcal{P}(Q), \Sigma, \delta', I, F')$, on per a tot $S \subseteq Q$ i $a \in Q$

1. $\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$
2. $S \in F' \Leftrightarrow S \cap F \neq \emptyset$

Ara tenim que per a tot $x \in \Sigma^*$,

$$
\begin{aligned}
x \in L(M) &\Leftrightarrow \exists q_0 \in I \;\; \delta(q_0, x) \cap F \neq \emptyset \\
&\Leftrightarrow \bigcup_{q_0 \in I} \delta(q_0, x) \cap F \neq \emptyset \\
&\Leftrightarrow \delta'(I, x) \cap F \neq \emptyset \\
&\Leftrightarrow \delta'(I, x) \in F' \\
&\Leftrightarrow x \in L(M').
\end{aligned}
$$

Corol·lari

Els llenguatges reconeguts per NFAs són regulars.

Així, la transformació és $M = (Q, \Sigma, \delta, I, F) \longrightarrow M' = (\mathcal{P}(Q), \Sigma, \delta', I, F')$, on per a tot $S \subseteq Q$ i $a \in Q$

1. $\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$
2. $S \in F' \Leftrightarrow S \cap F \neq \emptyset$

Ara tenim que per a tot $x \in \Sigma^*$,

$$\begin{aligned} x \in L(M) &\Leftrightarrow \exists q_0 \in I \ \ \delta(q_0, x) \cap F \neq \emptyset \\ &\Leftrightarrow \bigcup_{q_0 \in I} \delta(q_0, x) \cap F \neq \emptyset \\ &\Leftrightarrow \delta'(I, x) \cap F \neq \emptyset \\ &\Leftrightarrow \delta'(I, x) \in F' \\ &\Leftrightarrow x \in L(M'). \end{aligned}$$

Corol·lari

Els llenguatges reconeguts per NFAs són regulars.

Així, la transformació és $M = (Q, \Sigma, \delta, I, F) \longrightarrow M' = (\mathcal{P}(Q), \Sigma, \delta', I, F')$, on per a tot $S \subseteq Q$ i $a \in Q$

1. $\delta'(S, a) = \bigcup_{q \in S} \delta(q, a)$
2. $S \in F' \Leftrightarrow S \cap F \neq \emptyset$

Ara tenim que per a tot $x \in \Sigma^*$,

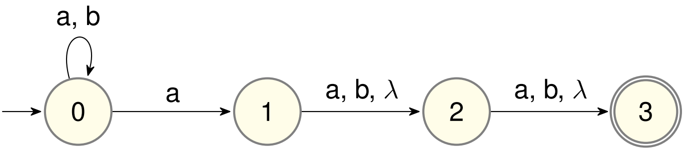
$$\begin{aligned} x \in L(M) &\Leftrightarrow \exists q_0 \in I \;\; \delta(q_0, x) \cap F \neq \emptyset \\ &\Leftrightarrow \bigcup_{q_0 \in I} \delta(q_0, x) \cap F \neq \emptyset \\ &\Leftrightarrow \delta'(I, x) \cap F \neq \emptyset \\ &\Leftrightarrow \delta'(I, x) \in F' \\ &\Leftrightarrow x \in L(M'). \end{aligned}$$

Corol·lari

Els llenguatges reconeguts per NFAs són regulars.

λ-transicions

El model indeterminista també permet tenir λ-transicions, és a dir, transicions que no consumeixen símbols d'entrada. Una λ-transició s'etiqueta amb λ en el diagrama d'estats.

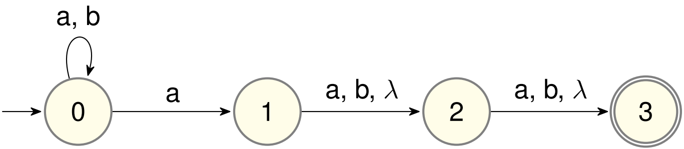


Qüestió

Quin és el llenguatge reconegut per l'autòmat anterior?

λ-transicions

El model indeterminista també permet tenir λ-transicions, és a dir, transicions que no consumeixen símbols d'entrada. Una λ-transició s'etiqueta amb λ en el diagrama d'estats.



Qüestió

Quin és el llenguatge reconegut per l'autòmat anterior?

λ-transicions

Definició

El model de NFAs amb λ-transicions s'anomena λ-NFA. La definició és la mateixa excepte que es permet fer una transició sense símbol d'entrada (equivalentment, amb λ com a entrada).

Sigui $\Sigma_\lambda = \Sigma \cup \{\lambda\}$. Un λ-NFA és un quíntuple $(Q, \Sigma, \delta, I, F)$, on:

- Q és un conjunt finit d'estats
- Σ és un alfabet,
- $\delta : Q \times \Sigma_\lambda \to \mathcal{P}(Q)$ és una funció total (de transició),
- $I \subseteq Q$ és el conjunt d'estats inicials i
- $F \subseteq Q$ és el conjunt d'estats acceptadors.

λ-transicions

Definició

El model de NFAs amb λ-transicions s'anomena λ-NFA. La definició és la mateixa excepte que es permet fer una transició sense símbol d'entrada (equivalentment, amb λ com a entrada).

Sigui $\Sigma_\lambda = \Sigma \cup \{\lambda\}$. Un λ-NFA és un quíntuple $(Q, \Sigma, \delta, I, F)$, on:

- Q és un conjunt finit d'estats
- Σ és un alfabet,
- $\delta : Q \times \Sigma_\lambda \to \mathcal{P}(Q)$ és una funció total (de transició),
- $I \subseteq Q$ és el conjunt d'estats inicials i
- $F \subseteq Q$ és el conjunt d'estats acceptadors.

Transformació de λ-NFA en NFA

Per construir un NFA a partir d'un λ-NFA es considera el λ-tancament dels estats. Si Q és el conjunt d'estats d'un λ-NFA i $p \in Q$, es defineix

$$\Lambda(p) = \{q \in Q \mid \text{ es pot arribar de } p \text{ a } q \text{ en un camí de } \lambda\text{-transicions}\},$$

Exemple: λ-tancament

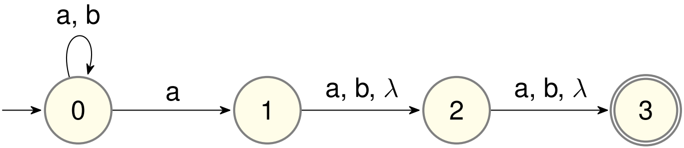
- $\Lambda(0) = \{0\}$
- $\Lambda(1) = \{1, 2, 3\}$
- $\Lambda(2) = \{2, 3\}$
- $\Lambda(3) = \{3\}$

Transformació de λ-NFA en NFA

Per construir un NFA a partir d'un λ-NFA es considera el λ-tancament dels estats. Si Q és el conjunt d'estats d'un λ-NFA i $p \in Q$, es defineix

$$\Lambda(p) = \{q \in Q \mid \text{ es pot arribar de } p \text{ a } q \text{ en un camí de } \lambda\text{-transicions}\},$$

Exemple: λ-tancament



- $\Lambda(0) = \{0\}$
- $\Lambda(1) = \{1, 2, 3\}$
- $\Lambda(2) = \{2, 3\}$
- $\Lambda(3) = \{3\}$

Equivalència entre λ-NFAs i NFAs

Teorema

Tot λ-NFA té un NFA equivalent.

Transformació

Donat un λ-NFA $M = (Q, \Sigma, \delta, I, F)$, definim l'NFA

$$M' = (Q, \Sigma, \delta', I, F'),$$

on per a tot $q \in Q$, $a \in \Sigma$

$$\delta'(p, a) = \bigcup_{q \in \Lambda(p)} \delta(q, a)$$

i F' conté els estats que tenen un camí fins a un estat de F a través de λ-transicions, és a dir,

$$F' = \{q \mid \Lambda(q) \cap F \neq \emptyset\}.$$

Equivalència entre λ-NFAs i NFAs

Teorema

Tot λ-NFA té un NFA equivalent.

Transformació

Donat un λ-NFA $M = (Q, \Sigma, \delta, I, F)$, definim l'NFA

$$M' = (Q, \Sigma, \delta', I, F'),$$

on per a tot $q \in Q$, $a \in \Sigma$

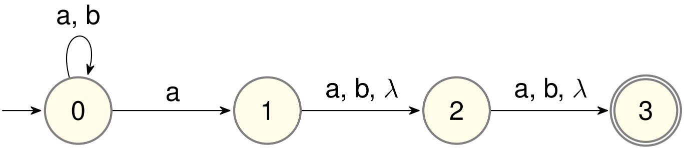
$$\delta'(p, a) = \bigcup_{q \in \Lambda(p)} \delta(q, a)$$

i F' conté els estats que tenen un camí fins a un estat de F a través de λ-transicions, és a dir,

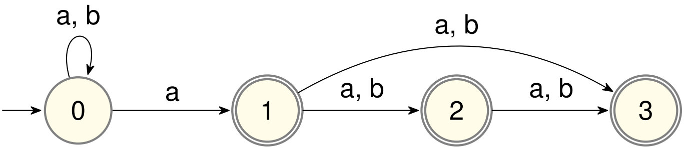
$$F' = \{q \mid \Lambda(q) \cap F \neq \emptyset\}.$$

Transformació de λ-NFA en NFA

Exemple



- $\Lambda(0) = \{0\}$, $\Lambda(1) = \{1, 2, 3\}$, $\Lambda(2) = \{2, 3\}$, $\Lambda(3) = \{3\}$



Autòmats finits

1 Concepte

2 Exemples

3 Definició formal

4 Llenguatge reconegut

5 Tancament per operacions 1

6 Indeterminisme

7 Tancament per operacions 2

El model de λ-NFA permet demostrar (de forma constructiva) el tancament dels regulars per totes les operacions que preserven la regularitat.

Com a exemples (sense demostració), veiem les construccions corresponents a la reunió, la concatenació i l'estrella de Kleene.

Reunió

REG és tancada per reunió.

En la figura següent es construeix un λ-NFA N que reconeix la reunió dels llenguatges dels λ-NFAs N_1 i N_2.

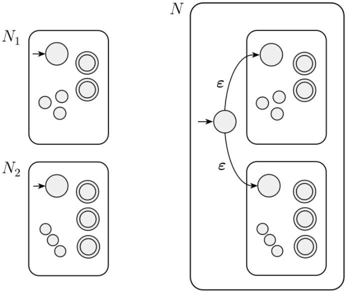


Figura: M. Sipser. *Introduction to the Theory of Computation*

Concatenació

REG és tancada per concatenació.

En la figura següent es construeix un λ-NFA N que reconeix la concatenació dels llenguatges dels λ-NFAs N_1 i N_2.

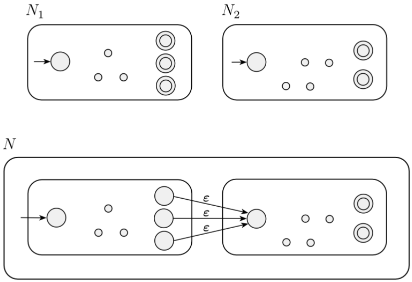


Figura: M. Sipser. *Introduction to the Theory of Computation*

Estrella de Kleene

REG és tancada per l'estrella de Kleene.

En la figura següent es construeix un λ-NFA N que reconeix la concatenació dels llenguatges dels λ-NFAs N_1 i N_2.

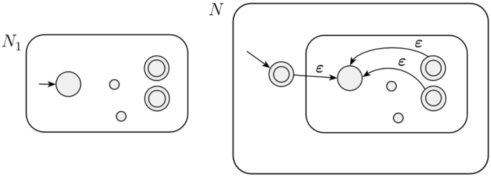


Figura: M. Sipser. *Introduction to the Theory of Computation*

Tancament positiu

REG és tancada pel tancament positiu.

Com que el tancament positiu d'un llenguatge L es pot expressar com

$$L^+ = L \cdot L^*,$$

si $L \in$ REG, llavors $L^* \in$ REG aplicant el tancament per l'estrella de Kleene, però llavors $L^+ \in$ REG aplicant el tancament per concatenació.

La construcció segueix la idea de l'estrella de Kleene però sense afegir el nou estat acceptador del requadre extrern en la figura anterior.