LIMBAJE FORMALE ȘI AUTOMATE – CURSUL 5

Descrierea limbajelor regulate cu ajutorul unei relații de echivalență de indice finit

Definiție. Fie $L \subseteq \Sigma^*$ un limbaj peste alfabetul Σ. Definim pe Σ^* următoarea relație

$$\forall x, y \in \Sigma^*,\ x \equiv_L y \Leftrightarrow (\forall z \in \Sigma^*, xz \in L \text{ dacă și numai dacă } yz \in L)$$

Aceasta este o relație de echivalență peste Σ^*, deoarece $\forall x, y, z \in \Sigma^*$ avem:

- $x \equiv_L x$ (idempotență)
- dacă $x \equiv_L y$ atunci $y \equiv_L x$ (simetrie)
- dacă $x \equiv_L y$ și $y \equiv_L z$ atunci $x \equiv_L z$ (tranzitivitate)

Totodată,

- (invarianța la dreapta la concatenare a relației $\equiv_L$) Dacă $x \equiv_L y$ atunci $\forall w \in \Sigma^*, xw \equiv_L yw$. Într-adevăr, dacă $x \equiv_L y$ atunci $\forall z \in \Sigma^*$, $xz \in L$ dacă și numai dacă $yz \in L$. Fie $u \in \Sigma^*$. Atunci pentru $z = wu$, avem $x(wu) \in L$ dacă și numai dacă $y(wu) \in L$, adică $(xw)u \in L$ dacă și numai dacă $(yw)u) \in L$, deci $xw \equiv_L yw$.

Definiție. Definim indicele unei relații de echivalență R peste Σ^*, notat cu $|\Sigma^* \backslash R|$, numărul claselor de echivalență ale relației R peste Σ^*.

Definiție. Pentru un automat finit determinist (prescurtat AFD), $A = (Q, \Sigma, \delta, q_0, F)$, definim relația $\equiv_A$ astfel:

$$\forall x, y \in \Sigma^*, x \equiv_A y \Leftrightarrow \delta(q_0, x) = \delta(q_0, y)$$

Evident, relația $\equiv_A$definită mai sus este de relație de echivalență și are indicele finit, deoarece A are un număr finit de stări. Pentru a arăta invarianța la dreapta la concatenare, fie $x \equiv_A y$ *și* $w \in \Sigma^*$. Avem $\delta(q_0, x) = \delta(q_0, y)$ și $\delta(q_0, xw) = \delta(\delta(q_0, x), w) = \delta(\delta(q_0, y), w) = \delta(q_0, yw)$, deci $xw \equiv_A yw$, adică $\equiv_A$ este invariantă la dreapta.

Teoremă. (Myhill-Nerode) Următoarele afirmații sunt echivalente:

1. L este un limbaj regulat;
2. L este reuniunea claselor de echivalență ale unei relații de echivalență invariantă la dreapta, de indice finit;

3. Relația $\equiv_L$ este de indice finit.

Demonstrație. $1 \Rightarrow 2$. Fie $L \subseteq \Sigma^*$ un limbaj regulat peste alfabetul Σ. Atunci există un AFD $A = (Q, \Sigma, \delta, q_0, F)$ cu $L(A) = L$. Putem presupune că A este complet (adică $\forall q \in Q, \forall a \in \Sigma, \delta(q, a)$ nevida, deci $\delta(q, a) \in \Sigma$) și, în plus, putem presupune că în A nu există stări inaccesibile (adică stări q pentru care nu există $w \in \Sigma^*$ astfel încât $\delta(q_0, w) = q$; vezi exercițiul de la finalul cursului). Considerăm relația de echivalență de indice finit, $\equiv_A$, definită prin $x \equiv_A y \Leftrightarrow \delta(q_0, x) = \delta(q_0, y)$. Atunci putem scrie:

$$L(A) = \bigcup_{q \in F} [q] = \{x \in \Sigma^* | \exists \mathrm{q} \in \mathrm{F}, \delta(q_0, x) = q\}$$

$2 \Rightarrow 3$. Fie R o relație care satisface 2, deci L este reuniunea claselor de echivalență ale R. Demonstrăm că R este o rafinare a lui $\equiv_L$, adică $xRy \Rightarrow x \equiv_L y$, ceea ce va implica că relația $\equiv_L$ are indicele finit. Fie xRy. Deoarece R este invarianta la dreapta, rezulta ca $\forall w \in \Sigma^*$, atunci $xwRyw$. Deoarece L este reuniunea unor clase de echivalență ale lui R rezulta ca xw și yw fac parte din aceeași clasă de echivalență, deci $xw \in L \Leftrightarrow yw \in L$, adica $x \equiv_L y$.

$3 \Rightarrow 1$. Știm că $\equiv_L$ are indicele finit și este invariantă la dreapta. Notăm cu $[w]$ clasa de echivalență a lui w, adică $[w] = \{x \in \Sigma^* | x \equiv_L w\}$. Fie $[\lambda], [x_1], \dots, [x_n]$ clasele de echivalență ale lui $\equiv_L, n \geq 0$. Observam ca daca $x \in L, y \in [x]$, atunci $y \in L$. Intr-adevar, $x \equiv_L y, \lambda \in \Sigma^*$, $\equiv_L$ invarianta la dreapta, deci $x \in L \Leftrightarrow y \in L$.

Construim AFD $A' = (Q', \Sigma, \delta', [\lambda], F')$ în felul următor:

$Q' = \{[\lambda], [x_1], \dots, [x_n]\}$,

$F' = \{[w] | w \in L\}$,

$\delta'([w], a) = [wa], [w] \in Q', a \in \Sigma$.

δ' este bine definita, deoarece dacă $v \in [w]$, atunci $\delta'([v], a) = [va]$ și deoarece $\equiv_L$ este invariantă la dreapta, $[va] = [wa]$, deci $\delta'([v], a) = [wa] = \delta'([w], a)$.

Avem $w \in L(A') \Leftrightarrow \delta'([\lambda], w) \in F' \Leftrightarrow [w] \in F' \Leftrightarrow w \in L$, deci $L = L(A')$, adică L este regulat.

Minimizarea automatelor finite deterministe

Teoremă. Automatul finit determinist cu număr minim de stări care acceptă limbajul L este unic, făcând abstracție de un izomorfism, și este dat de automatul A' construit ca în teorema anterioară.

Demonstrație. Fie AFD $A = (Q, \Sigma, \delta, q_0, F)$ cu număr minim de stări, $n = |Q|$, care recunoaște L, $L(A) = L$. Din Teorema anterioară am văzut că $\equiv_A$ este o relație de echivalență de indice finit care rafinează $\equiv_L$. Aceasta înseamnă că $|\Sigma^*/\equiv_A| \geq [\Sigma^*/\equiv_L]$, iar $|\Sigma^*/\equiv_A| \leq n$, deoarece numarul claselor de echivalență ale lui $\equiv_A$ nu poate depăși numărul stărilor din Q . Pe de altă parte, AFD $A' = (Q', \Sigma, \delta', [\lambda], F')$ construit ca in teorema Myhill-Nerode recunoaste pe L si $|Q'| = |\Sigma^*/\equiv_L|$, și cum A are un număr minim de stări, rezultă că $|Q'| = |\Sigma^*/\equiv_L| = n = |Q| = |\Sigma^*/\equiv_A|$. Consideram izomorfismul $f: Q \to Q'$ definit prin $f(q) = [w] \Leftrightarrow \delta(q_0, w) = q$. Izomorfismul f este bine definit: pentru $v \in [w], v \equiv_A w \Leftrightarrow \delta(q_0, w) = \delta(q_0, v) \Leftrightarrow f(q) = [v]$.

Algoritm pentru minimizarea stărilor unui AFD

Pentru AFD $A = (Q, \Sigma, \delta, q_0, F)$ considerăm următoarea relație de echivalență definită pe Q: $p \equiv q \Leftrightarrow (\forall w \in \Sigma^*, \delta(p, w) \in F \Leftrightarrow \delta(q, w) \in F$. Observăm că $\equiv$ este o relație pentru care există o bijecție f de la clasele lui $\equiv$ la Q' construită în teorema Myhill-Nerode, definită prin $f([q]) = [w] \Leftrightarrow \delta(q_0, w) \in [q]$. Deci putem calcula $A' = (Q', \Sigma, \delta', [\lambda], F')$ construit ca in teorema Myhill-Nerode dacă calculăm clasele lui $\equiv$.

INPUT AFD $A = (Q, \Sigma, \delta, q_0, F)$ complet, fără stări inaccesibile
OUTPUT AFD $A' = (Q', \Sigma, \delta', [\lambda], F')$ construit ca in teorema Myhill-Nerode cu număr minim de stări, $Q' \subseteq 2^Q$
Algoritmul marchează perechi (neordonate) de stări (p, q). O pereche (p, q) este marcată îndată ce se constată că p și q nu sunt echivalente.

1. Inițializează o matrice X[p,q] cu 0, pentru orice p,q∈Q
2. Marchează X[p,q] cu 1 dacă p∈F și q∉ F sau invers
3. **do{**
    if (X[p,q]==0 **and** (există a∈Σ, X(δ(p,a),δ(q,a))=1)

```
        X[p,q]← 1
      }while (se fac modificări ale lui X)
4. Pentru fiecare q∈ Q, se construiește clasa de echivalență [q] cu ajutorul lui X.
  4.1 Q'← {[q0]}, [q0] stare nemarcata
  4.2 while (exista T⊆ Q stare nemarcata)
       { marcheaza T;
         for (fiecare a∈Σ)
           {V← [∪p∈T δ(p,a)];
            δ'(T,a)←V;
            if (V∉ Q') adauga V la Q' ca stare nemarcata;
           }
         }
5. F'← {V∈ Q'| V∩F≠ ∅}
```

Observații.

- $X[p,q]=0$ dacă și numai dacă p și q sunt echivalente
- Dacă $X[p,q]$ devine 1 la pasul 2, atunci p și q nu sunt echivalente
- La pasul 3 se poate verifica o pereche (p,q) de mai multe ori
- Algoritmul este de ordinul $O(|\Sigma|\times|Q|^2)$
- Există și o variantă mai bună, $O(|\Sigma|\times|Q|\times\log(|Q|))$

Exemplu. Fie AFD dat de următoarea funcție de tranziție

	a	b
0	1	2
$1F$	3	4
$2F$	4	3
3	5	5
4	5	5
$5F$	5	5

Automatul are stările {0,1,2,3,4,5}, iar cele finale au fost marcate cu *F*.
La pasul 1, toate perechile sunt nemarcate ($X[i,j]=0$, $i\le j$)

0					
0	**1**				
0	0	**2**			
0	0	0	**3**		
0	0	0	0	**4**	
0	0	0	0	0	**5**

Dupa pasul 2, sunt marcate toate perechile constând dintr-o stare finală și una nefinală

0					
1	**1**				
1	0	**2**			
0	1	1	**3**		
0	1	1	0	**4**	
1	0	0	1	1	**5**

Acum să urmărim perechea nemarcată (0,3). Cu 'a', 0 și 3 merg în 1 și 5, respectiv. Perechea (1,5) nu este marcată, deci nu marcăm (0,3), cel puțin deocamdată. Apoi urmărim perechile (0,4) și (1,2) și găsim că nu le putem marca, din același motiv. Dar pentru (1,5) și simbolul de intrare 'a', (1,5)→ (3,5), iar (3,5) este marcată, deci marcăm (1,5). Similar, pentru simbolul 'a' (2,5)→ (4,5) care este marcată, deci marcăm (2,5). Cu 'a' și 'b', (3,4)→ (5,5), care nu este niciodată marcată, deci nu marcăm (3,4). După prima trecere de la pasul 3 avem:

0					
1	**1**				
1	0	**2**			
0	1	1	**3**		
0	1	1	0	**4**	
1	1	1	1	1	**5**

Acum facem o nouă trecere prin matricea X. Ca și mai înainte, (1,2)→ (3,4) și (3,4)→ (5,5), atât cu 'a' cât și cu 'b', și nici (3,4) nici (5,5) nu sunt marcate, deci nu mai avem marcări noi. Ramân (1,2) si (3,4) nemarcate, ceea ce indică că fac parte din aceleași clase de echivalență.

Obținem: [0]={0}=A, [1]=[2]={1,2}=B, [3]=[4]={3,4}=C, [5]={5}=D, iar funcția de tranziție pentru noul automat va fi

	a	b
A	B	B
BF	C	C
C	D	D
DF	D	D

Unde 'F' indică stările finale, iar A este starea inițială.

Exerciții

1) Fie un AFD $A = (Q, \Sigma, \delta, q_0, F)$. Descrieți în pseudocod un algoritm care depistează stările inaccesibile q∈ Q pentru care $\nexists w \in \Sigma^*$ astfel încât $\delta(q_0, w) = q$. Totodată, algoritmul va elimina toate tranzițiile care implică astfel de stări inaccesibile și apoi va elimina stările inaccesibile înseși din Q.

2) Fie un AFD $A = (Q, \Sigma, \delta, q_0, F)$ necomplet (adică $\exists a \in \Sigma, \exists \mathrm{q} \in \mathrm{Q}, \delta(\mathrm{q}, \mathrm{a}) = \emptyset$). Descrieți în pseudocod un algoritm prin care se obtine un AFD complet A'cu $L(A) = L(A')$.