CURS

SISTEME TRANZITIONALE și EXPRESII REGULATE

Sistemele tranzitionale sunt o extindere a automatelor finite ce admit mai multe stări inițiale (ni loc de s_0) și tranziții spontane (adică schimbarea stării fără deplasarea capului de citire (sau citirea de pe b.i. a lui λ). Vom arăta că puterea de recunoaștere a sistemelor tranzitionale este identică cu a AF uzuale.

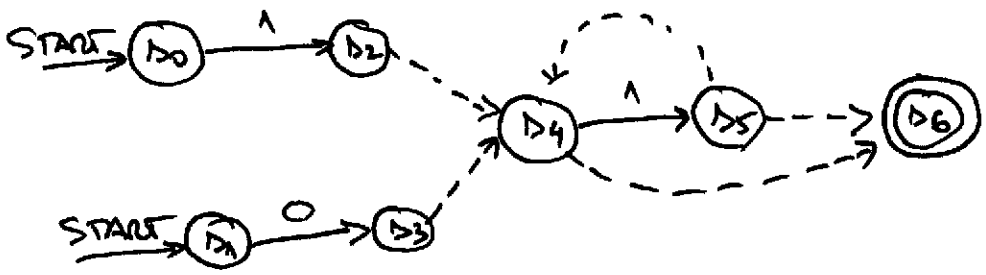
Definiția formală: Un sistem tranzitional este

$$ST = (S, I, f, S_0, S_f, \delta)$$

unde
S, I, f, S_f sunt identice cu definitiile AF
$S_0 \subset S$ - submultimea de stări inițiale
$\delta \subset S \times S$ - relația de tranzitie spontană
Spunem că $(s, s') \in \delta \iff$ ST aflat în starea $s \in S$ poate trece în starea s' prin citirea lui λ

Diagrama de stări ST: La diagrama obișnuită a unui AF adaug tranzitiile spontane definite de relația δ. Grafic se folosesc arce punctate de la s la s' (uneori se notează λ pe arc!)

Ex:



Aici $S_0 = \{s_0, s_1\}$

$\delta = \{(s_2, s_4), (s_3, s_4), (s_5, s_4), (s_5, s_6), (s_4, s_6)\}$

Se defineste un pas de functionare pt ST astfel.

Fie $i \in I \cup \{\lambda\}$.

Def: Spunem că ST evoluează direct din starea s în starea s' și notăm $s \vdash^{i} s'$ dacă:

(1) $i = \lambda$ și $(s, s') \in \delta^*$

(2) $i \in I$ și $\exists\, s_1, s_2 \in S$ a.î. $(s, s_1) \in \delta^*$, $f(s_1, i) \ni s_2$, $(s_2, s') \in \delta^*$

Obs: un ST poate evolua direct prin citirea lui λ (sir de tranziții spontane), cazul (1) ceea ce revine la parcurgerea unei traiectorii punctate de la s la s' sau $s \equiv s'$

Ptr. cazul (2) ST poate evolua direct prin citirea unui simbol de pe b.i și avansează o poziție dreapta, ceea ce înseamnă că din starea s se parcurg eventual arce punctate până la originea unui arc plin, se citește sb. de pe b.i., apoi se parcurg eventual arce punctate până la s'

Ptr. cazul ST din exemplu avem

$s_0 \vdash^{\wedge} s_2$ $\quad s_2 \vdash^{\lambda} s_4$

$s_0 \vdash^{\wedge} s_4$ $\quad s_2 \vdash^{\lambda} s_6$

$s_0 \vdash^{\wedge} s_6$ $\quad s_2 \vdash^{\wedge} s_6$ $\quad$ s.a.m.d.

Evoluția directă se poate extinde la cuvinte $p = i_1 \dots i_n \in I^*$ astfel (închiderea tranzitivă!)

$s \vdash^{p} s' \overset{def}{\Leftrightarrow} \exists\, s_1, s_2, \dots, s \in S$ a.î.

$s \vdash^{i_1} s_1 \vdash^{i_2} s_2 \vdash \dots \vdash^{i_n} s'$

Limbajul recunoscut de un sistem tranzitional se defineste astfel $L(ST) = \{p \in I^* \mid \exists s \in S_0 \text{ a.i. } s \overset{p}{\longmapsto} s' \text{ si } s' \in S_f\}$

TEOREMĂ: $\mathcal{R} = \mathcal{L}_{ST}$

#

Evident un AF este un caz particular de ST.
Deci $\mathcal{R} \subset \mathcal{L}_{ST}$ evident.
Pentru incluziunea inversă $\mathcal{L}_{ST} \subset \mathcal{R}$ se construieşte un AF determinist echivalent astfel:
Dacă $ST = (S, I, f, S_0, S_f, \delta)$ atunci

$AF = (\mathcal{P}(S), I, f', S_0, S_f')$ unde

$f'(Z, i) = \{s \in S \mid \exists s' \in Z \text{ a.i. } s' \overset{i}{\longmapsto} s\}$

$S_f' = \{Z \in \mathcal{P}(S) \mid Z \cap S_f \neq \phi\}$

Rămâne să arătăm corectitudinea construcţiei, adică

$$L(AF) = L(ST)$$

($\subseteq$) Fie $p = i_1 \dots i_n \in L(ST)$. Există evoluţia

(a) $s_0 \overset{i_1}{\longmapsto} s_1 \overset{i_2}{\longmapsto} \dots \overset{i_n}{\longmapsto} s_n \in S_f$, $s_0 \in S_0$.

Putem construi următoarea traiectorie în AF

(b) $S_0 \overset{i_1}{\Longrightarrow} Z_1 \xrightarrow{i_2} \dots \xrightarrow{i_n} Z_n$,

unde $Z_1 = f'(S_0, i_1)$

$Z_2 = f'(Z_1, i_2)$

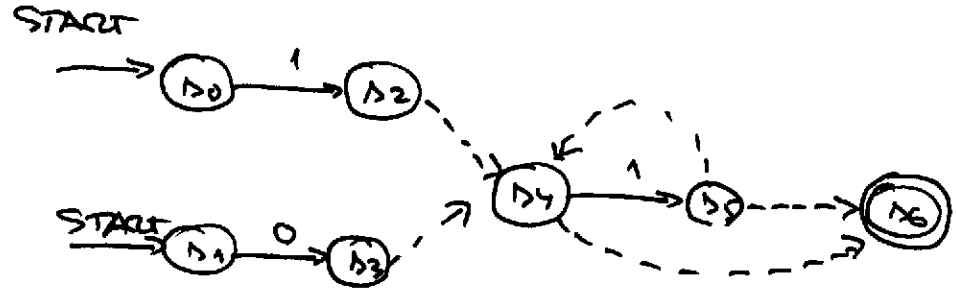
$\dots$

$Z_n = f'(Z_{n-1}, i_n)$

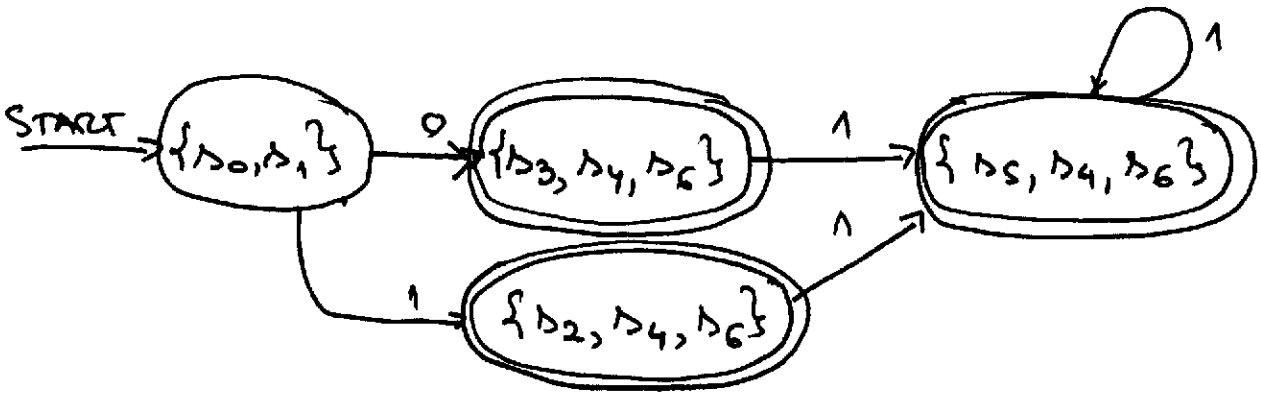
Să observăm că $s_0 \in S_0$, $s_1 \in Z_1 = f'(S_0, i_1)$, $\dots$, $s_n \in Z_n$
iar $s_n \in S_f \Rightarrow Z_n \cap S_f \neq \phi$, adică $Z_n \in S_f'$
adică $p \in L(AF)$.

($\supseteq$) Raţionamentul în sens invers.

Pth. ST din exemplu



AF echivalent este



Deci $L(ST) = L(AF) = \{0 1^n \mid n \geq 0\} \cup \{1^n \mid n \geq 1\}$

$= \{a 1^n \mid a \in \{0,1\}, n \geq 0\}$

EXPRESII REGULATE

notație pentru limbajele regulate (sau „șablon")

Fie V un alfabet.

DEF 1: Se numește expresie regulată peste alfabetul V un cuvânt peste alfabetul extins $V \cup \{\cdot, *, |\} \cup \{(,)\} \cup \{\phi\}$ obținut prin aplicarea de un număr finit de ori a următoarelor reguli:

(1) λ și ϕ sunt expresii regulate;

(2) a este e.r. $\forall a \in V$;

(3) dacă R și S sunt e.r. atunci $(R|S)$, $(R \cdot S)$ și (R^*) sunt e.r. peste V;

Obs: 1) $|, \cdot, *$ sunt numiți operatori sau, produs (sau și), iterație

$(,)$ – parantezele sunt folosite pt a pune în evidență ordinea aplicării operatorilor

ϕ – simbol folosit ca notație în multimea vidă.

2) Definiția recursivă este similară cu definirea expr. aritm/logice

3) λ, ϕ, a sunt expresii elementare, ca și constantele/variabilele din e.a.

Ex: $V = \{0,1\}$. Exemple de e.r. peste V

0 este e.r. (regula 2)

λ este e.r. (regula 1)

$(0|1)$ este e.r. (0 e.r, 1 e.r regula 2 apoi regula 3)

$((0^*) \cdot 1)$ este e.r. (0 e.r $\Rightarrow (0^*)$ e.r 1 e.r $\overset{\text{reg 3}}{\Rightarrow} ((0^*) \cdot 1)$ e.r)

Se pot evita parantezele considerând prioritati pentru operatori $p(*) = 3$, $p(\cdot) = 2$, $p(|) = 1$, în mod similar cu expresiile aritmetice. În plus, nu se scrie explicit operatorul produs $\cdot$, ca în cazul expresiilor algebrice.

$((0^*) \cdot 1) \overset{\text{not}}{=} 0^*1$

$(0|1) = 0|1$

Expresiile regulate sunt asociate unor multimi de cuvinte astfel.

(1) λ notează multimea $\{\lambda\}$

(1') ϕ notează multimea vidă ϕ

(2) a notează multimea $\{a\}$

(3) Dacă R și S sunt expresii regulate ce notează multimile L_R și L_S atunci

3.1. e.r.$(R|S)$ notează multimea $L_R \cup L_S$

3.2. e.r. $(R \cdot S) \overset{not}{=} L_R \cdot L_S$

3.3 e.r $(R^*) \overset{not}{=} (L_R)^*$

Obs: Operatorii $|, \cdot, *$ corespund operațiilor regulate asupra limbajelor

Prin abuz de limbaj folosim notația cu e.r. pt limbaje

Exemplul 1: $V = \{0,1\}$

$0 \overset{not}{=} \{0\}$

$0^* \overset{not}{=} \{0\}^* = \{0^n \mid n \geq 0\}$

$10^* \overset{not}{=} \{10^n \mid n \geq 0\}$

$(0|1)^* \overset{not}{=} \{0,1\}^*$

$(0|1)^*1 \overset{not}{=} \{w1 \mid w \in \{0,1\}^*\} = \{w \in \{0,1\}^* \mid w$ se termină cu $1\}$

Exemplul 2: $V = \{a,b,c\}$

$\phi \overset{not}{=} \phi$

$a^*b \overset{not}{=} \{a^n b \mid n \geq 0\}$

$(a|b|c)^* b (a|b|c)^* \overset{not}{=} \{wbw' \mid w,w' \in \{a,b,c\}^*\}$

$(a|b|c)^* a (a|b|c) \overset{not}{=} \{w \in \{a,b,c\}^* \mid w$ are în penultima poziție litera $a\}$

$(a|b|c)^* baba (a|b|c)^* \overset{not}{=} \{w \in \{a,b,c\}^* \mid w$ conține subcuvântul baba$\}$

TEOREMĂ Expresiile regulate notează limbaje regulate.

//

Se construiește un ST ce recunoaște limbajul notat de o expresie regulată. Algoritmul poartă numele ALGORITMUL LUI THOMPSON.

Se construiesc sisteme tranziționale asociate aplicațiilor regulilor ce definesc construcția expresiei regulate. ST au o singură stare inițială și una finală.

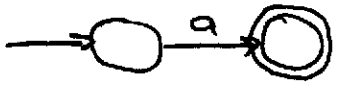
Fie V alfabetul peste care construim expresia.

Următoarele ST recunosc e.r. elementare

(1) 

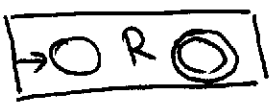
λ (1') 

ϕ

(2) 

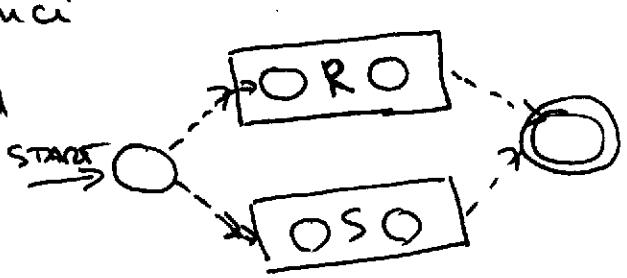
recunoaște $\{a\}$ $a \overset{not}{=} \{a\}$

(3) Dacă R și S sunt e.r. și

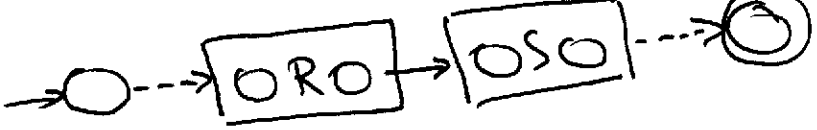


Sunt ST ce recunosc limbajele L_R și L_S

atunci

3.1 

recunoaște $R|S \overset{not}{=} L_R \cup L_S$

3.2 

$R \cdot S$

3.3. 

R^*

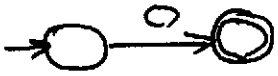
Combinările 3.1, 3.2. 3.3 se mai numesc legarea în paralel, serie respectiv secvențială a ST.

Deci $R = L_{ST}$

Exemplu: Considerăm $V=\{0,1\}$ și

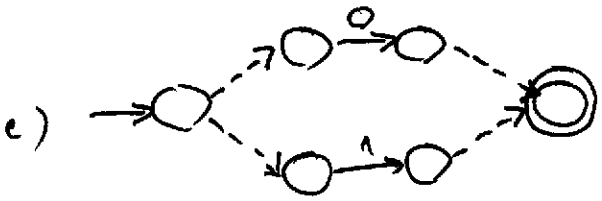
$(0|1)^*1 \overset{not}{=} L=\{w\in\{0,1\}^* \mid w \text{ se termină cu } 1\}$

Avem urm. ST elementare corespunzătoare literelor 0 și 1

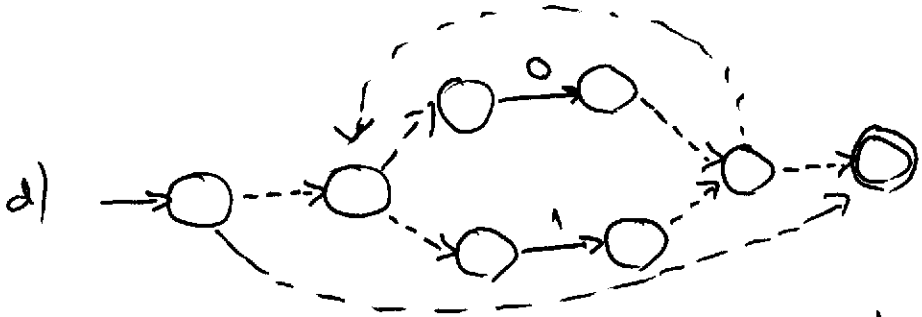
a) 

b) 

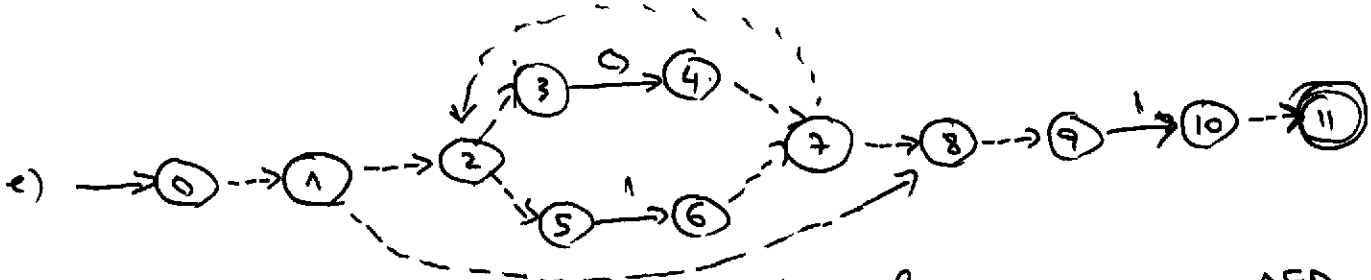
Ptr. 0|1 legăm cu paralel diagramele a) și b)

c) 

Ptr $(0|1)^*$ legăm cu săgeți circuit diagrama c)

d) 

Ptr $(0|1)^*1$ legăm în serie diagramele d) și b)

e) 

Sistemul tranzitiv se transformă ușor în AFD.