

MODELI RAČUNARSTVA

(Jezični procesori 1)

dr.sc. Julije Ožegović, izv.prof.

Predavanja

Laboratorijske Vježbe

MODELI RAČUNARSTVA - JEZIČNI PROCESORI 1

Siniša Srbljić, Sveučilište u Zagrebu

1. UVOD

2. REGULARNI JEZICI

3. KONTEKSTNO NEOVISNI JEZICI

4. REKURZIVNO PREBROJIVI JEZICI

5. KONTEKSTNO OVISNI JEZICI

6. RAZREDBA (TAKSONOMIJA) JEZIKA, AUTOMATA I GRAMATIKA

1. UVOD



1.1. ZNAKOVLJE I OZNAKE

1.2. PRIMJER FORMALNOG JEZIKA, AUTOMATA I GRAMATIKE

1. UVOD

- RAČUNALO
rješenje problema zahtjeva izgradnju algoritma
- JEZICI: C, C++, Java
koriste se za izgradnju algoritma
- JEZIČNI PROCESORI
prevode jezik izgradnje algoritma u jezik stroja

JEZIČNI PROCESOR

- PROBLEM X, RAČUNALO R
programski jezik L_{ciljni} ugrađen je u R
(jezik stroja)
- ALGORITAM A RJEŠAVA X
zapisan korištenjem jezika $L_{izvorni}$
- JEZIČNI PROCESOR
prevodi zapis iz L_i u L_c

JEZIČNI PROCESOR

- JEZIČNI PROCESOR JE PROGRAM
čita izvorni program i prevodi ga u ciljni program

- ZASNOVAN JE NA TRI JEZIKA

izvorni jezik L_i

ciljni jezik L_c

jezik izgradnje L_g

$$JP_{L_g}^{L_i \rightarrow L_c}$$

- NAJČEŠĆE

$$L_c = L_g$$

PROGRAM

- PROGRAM JE NIZ ZNAKOVA

```
import java.awt.*;public class Example{public static void main(String args[]){}}
```

- može se čitkije pisati (strukturirano)

```
import java.awt.*;
public class Example
{
    public static void main(String args[])
    {
    }
}
```

- primjer p1 **je ispravan** u skupu programa L_{java}

PROGRAM

- PROGRAM MOŽE BITI NEISPRAVAN

```
import java.awt.*;public class Example{public static void main(String args[]){}}
```

- može se čitkije pisati (strukturirano)

```
import java.awt.*;
public class Example
{
    public static void main(String args[])
    {
        {{
        }}
    }
}
```

- primjer p2 **nije ispravan** u skupu programa L_{java}

PROGRAM

- NEISPRAVAN PROGRAM p2
je član skupa svih programa L_{svi}
- ODNOS L_{svi} I L_{java}
 L_{java} je **pravi podskup** jezika L_{svi}

JEZIK

- JEZIK JE DEFINIRAN SKUPOM SVIH NIZOVA
 L_{java} sadrži sve ispravne programe
napisane u jeziku Java
- SKUP L_{java}
definira jezik Java i formalna je specifikacija
jezika Java

JEZIČNI PROCESOR

- OBZIROM NA FORMALNU DEFINICIJU JEZIKA jezični procesor:
 - prevodi program p_x u jeziku L_i (p_x u skupu L_i) u program p_y u jeziku L_c (p_y u skupu L_c)
- OKOSNICU ČINE DVA PROCESA
 - **prihvaćanje** izvornog programa
 - **generiranje** ciljnog programa

JEZIČNI PROCESOR

- PRIHVAĆANJE IZVORNOG PROGRAMA
 - jezični procesor čita znak po znak program p_x
 - ako je $p_x \in L_i$, dojavljuje se da je p_x ispravan
 - ako je $p_x \notin L_i$, dojavljuje se da je p_x neispravan, daljnji proces prevođenja se prekida, ispisuju se pogreške
- FORMALNI AUTOMAT M
 - gradi se za potrebe prihvatanja programa jezika L_i
 - to je matematički model automata - programa

JEZIČNI PROCESOR

- GENERIRANJE CILJNOG PROGRAMA
 - pokreće se ako je $p_x \in L_i$
 - generira se znak po znak ciljni program p_y u jeziku L_c
 - tako da je p_y prijevod izvornog programa p_x
- FORMALNA GRAMATIKA G
 - gradi se za potrebe generiranja programa jezika L_c
 - to je matematički model automata - programa
 - generira samo ispravne programe u L_c
 - generira p_y koji je upravo prijevod p_x

JEZIČNI PROCESOR

- STVARNI PROCES PREVOĐENJA
 - odvija se u više razina
 - svaka razina obavlja dvostupanjski proces
 - prvo se prihvaća program generiran na prethodnoj razini
 - nakon toga se generira program prema narednoj razini
 - početni proces prihvaća izvorni p_x ,
a završni generira ciljni p_y
- DVIJE FAZE S UKUPNO PET RAZINA
 - **analiza** u DVIJE razine
 - **sinteza** u TRI razine

JEZIČNI PROCESOR

- FAZA ANALIZE
 - prva razina: **leksička analiza**
 - druga razina: **sintaktička i semantička analiza**
generiranje **višeg međukoda**
- FAZA SINTEZE
 - treća razina: prevođenje višeg u **srednji međukod**
 - četvrta razina: prevođenje srednjeg u **niži međukod**
 - peta razina: prevođenje nižeg međukoda u **ciljni program**

JEZIČNI PROCESOR

- FAZA ANALIZE
 - odvija se u dvije razine
 - prva razina: **leksička analiza**
 - druga razina: **sintaktička i semantička analiza**
- LEKSIČKA ANALIZA
 - grupira znakove u osnovne elemente jezika
 - **leksičke jedinice** kao nizovi znakova su varijable, konstante, ključne riječi, operatori, pravopisni znakovi
 - **leksička pravila** određuju skup svih pravilno napisanih leksičkih jedinica, može biti beskonačan

JEZIČNI PROCESOR

- LEKSIČKI ANALIZATOR
 - formalni automat koji prihvata leksičke jedinice
 - prepoznaje članove skupa jedinice, a to je zapravo **jezik** leksičkih jedinice
 - nakon provjere leksičkih jedinice, zamjenjuju se **jedinstvenim znakovima**
- PRIMJER LEKSIČKE ANALIZE
 - program $PPP = \text{Jura}[i][j+7+24];$
 - prevodi se na $V \leftarrow V[V][V+K+K];$
 - svi ostali podaci spremaju se u **tablicu znakova**

JEZIČNI PROCESOR

- SINTAKSNA I SEMANTIČKA ANALIZA
 - sintaksnom analizom prihvaća se niz jedinstvenih znakova na osnovu **sintaksnih pravila**
 - semantičkom analizom generira se **viši međukod**
- SINTAKSNA PRAVILA DEFINIRAJU
 - način gradnje izraza od leksičkih jedinki (aritmetika)
 - način gradnje naredbi (pridruživanja, grananja)
 - način gradnje blokova naredbi i strukturu programa
 - jezik kao skup svih dozvoljenih nizova leksičkih jedinki
 - gradi se formalni automat - **sintaksni analizator**

JEZIČNI PROCESOR

- SEMANTIČKA PRAVILA
 - provjeravaju se prije generiranja se višeg međukoda
 - povezuju ponašanje računala s izvođenjem programa (npr. dodjela tipa varijable)
 - određuje skup dozvoljenih značenja (npr. provjera tipa varijable)
- PREVOĐENJE U VIŠI MEĐUKOD
 - izračunaju se konstantne vrijednosti
 - pojednostavni se struktura naredbe
 - koriste se deklaracije varijabli

JEZIČNI PROCESOR

- PRIMJER PREVOĐENJA U VIŠI MEĐUKOD
 - na osnovu deklaracije varijabli:
int i, j;
int PPP;
int Jura[][] = new int[20][100]
 - i, j i PPP su cjelobrojne varijable,
Jura je dvodimenzionalno polje
 - dobije se viši međukod: **PPP←Jura[i, j+31]**
 - time završava faza analize, slijedi faza sinteze

JEZIČNI PROCESOR

- FAZA SINTEZE

- odvija se u tri razine
- treća razina: prevođenje višeg u **srednji međukod**
- četvrta razina: prevođenje srednjeg u **niži međukod**
- peta razina: prevođenje nižeg međukoda u **ciljni program**

JEZIČNI PROCESOR

- PREVOĐENJE U SREDNJI MEĐUKOD

- pretvoriti složene strukture i naredbe u **osnovne**
- npr. pretvoriti dvodimenzionalno polje u jednodimenzionalno s duljinom riječi 4 okteta
korištenjem: `Jura[i, j+31]` i
`int Jura[][] = new int[20][100]`
dobije se indeks: $(i*100)+(j+31)$
- množenjem s 4 i dodavanjem početne adrese polja dobije se adresa traženog podatka u memoriji

JEZIČNI PROCESOR

- PRIMJER PREVOĐENJA U
SREDNJI MEĐUKOD

$v1 \leftarrow i * 100$

$v2 \leftarrow j + 31$

$v3 \leftarrow v1 + v2$

$v4 \leftarrow v3 * 4$

$v5 \leftarrow \text{AdrJura}$

$v6 \leftarrow v5 + v4$

$\text{PPP} \leftarrow @v6$

@aaa znači čitanje sa adrese aaa

JEZIČNI PROCESOR

- PREVOĐENJE U NIŽI MEĐUKOD
 - koriste se simbolički registri i stvarne adrese varijabli
 - varijable i i j spremaju se u prvih 8 okteta polja

- PRIMJER PREVOĐENJA U NIŽI MEĐUKOD

$r1 \leftarrow [\text{Početak}]$

$r2 \leftarrow r1 * 100$

$r3 \leftarrow [\text{Početak} + 4]$

$r4 \leftarrow r3 + 31$

$r5 \leftarrow r4 + r2$

$[aaa]$ znači čitanje sa adrese aaa

$r6 \leftarrow r5 * 4$

$r7 \leftarrow \text{Početak} + 8$

$r8 \leftarrow [r7 + r6]$

1.1. Znakovlje i oznake

ZNAK I ABECEDA

- **znak** je elementarni simbol, od kojeg se grade riječi
- **abeceda** je skup znakova, npr. $B = \{0, 1\}$

NIZ

- **niz** je konačni slijed znakova abecede postavljenih jedan iza drugog, npr. $w = 1011$
- **duljina niza** jednaka je broju njegovih znakova
npr. $|w| = 4$, u nizu w u 4 znaka abecede $B = \{0, 1\}$
- **prazni niz** ε nema znakova, $|\varepsilon| = 0$, neutralni je element nadovezivanja: $w\varepsilon = \varepsilon w = w$

Znakovlje i oznake

- NADOVEZIVANJE NIZOVA

- označava se potencijom: $w = yy = y^2$
- vrijedi: $w^0 = \varepsilon$; $w^i = w^{i-1}w$; $w^1 = w^0w = \varepsilon w = w$

- DIJELOVI NIZOVA

- prefiks se dobije odbacivanjem 0, 1 ili više posljednjih znakova, npr. banana
- sufiks se dobije odbacivanjem 0, 1 ili više početnih znakova, npr. banana
- podniz se dobije odbacivanjem sufiksa i prefiksa, npr. banana
- pravi prefiks, sufiks i podniz su neprazni nizovi
- podslijed se dobije odbacivanjem 0, 1 ili više ne nužno uzastopnih znakova, npr. banana

Znakovlje i oznake

- JEZIK

- je skup nizova nad abecedom $B=\{0,1\}$: $L_1=\{\}$, $L_2=\{\varepsilon\}$,
 $L_3 = \{00, 11, 0011, 0101, 1010, 1100, \dots\}$
 L_3 nije konačan
- jezik koji sadrži sve nizove nad abecedom Σ je Σ^* :
 $B^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100 \dots\}$

Znakovlje i oznake

- OPERACIJE NAD JEZICIMA

- **unija** L i N : $L \cup N = \{w \mid w \in L \vee w \in N\}$
- **presjek** L i N $L \cap N = \{w \mid w \in L \wedge w \in N\}$
- **razlika** L i N $L - N = L \setminus N = \{w \mid w \in L \wedge w \notin N\}$
- **nadovezivanje** L i N $LN = \{xy \mid x \in L \wedge y \in N\}$
- **kartezijski produkt** $L \times N = \{(x,y) \mid x \in L \wedge y \in N\}$
- **partitivni skup** 2^L je skup svih podskupova od L
- **Kleeneov operator** $*$ $L^* = \cup L^i; i=0 \dots \infty$
- **Kleeneov operator** $+$ $L^+ = \cup L^i; i=1 \dots \infty$
- **komplement** L $L^c = \{w \mid w \notin L\}$
- **nadovezivanje** L $LL = L^2$

Znakovlje i oznake

- BESKONAČNI SKUPOVI

- **kardinalni broj** je broj članova nekog skupa
- dva skupa imaju iste kardinalne brojeve ako postoji **bijekcija** među njihovim elementima
- skup i pravi podskup imaju **različite** kardinalne brojeve
- beskonačni skup i pravi podskup ne moraju imati nužno različite kardinalne brojeve, npr. $f(2i)=i$
- beskonačni skup je **prebrojivo beskonačan** ako postoji bijekcija na skup prirodnih brojeva \mathbb{N} (to su \mathbb{Z} i \mathbb{Q})
- skup realnih brojeva je **neprebrojivo beskonačan**

Znakovlje i oznake

- GRAF

- **graf** $G = (V, E)$ čini
 - konačni skup čvorova V i
 - skup parova čvorova E
- **put grafa** je niz čvorova $v_1, v_2, v_3, \dots, v_k$ ($k \geq 1$) za koji vrijedi $(v_i, v_{i+1}) \in E$, $1 \leq i \leq k$
- **duljina puta** je $k-1$

- USMJERENI GRAF

- grane usmjerenog grafa su **uređeni parovi** $v_1 \rightarrow v_2$
- za put usmjerenog grafa vrijedi $(v_i \rightarrow v_{i+1}) \in E$
- razlikujemo obične i neposredne prethodnike i sljedbenike

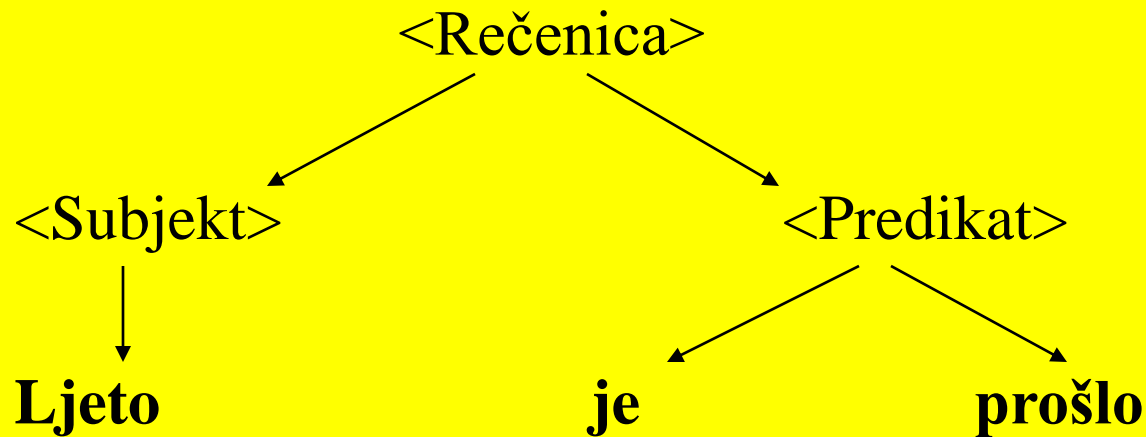
Znakovlje i oznake

- STABLO JE USMJERENI GRAF
 - **korijen** je čvor koji nema prethodnika
 - od korijena vodi put do svih ostalih čvorova
 - bilo koji čvor osim korijena ima točno jednog neposrednog prethodnika **roditelja**
 - ostali prethodnici su **preci**
 - neposredni sljedbenik je **dijete**
 - ostali sljedbenici su **potomci**
 - čvor bez djece je **list**, svi ostali čvorovi su **unutrašnji**

Znakovlje i oznake

- PRIMJER STABLA

- raščlamba rečenice “Ljeto je prošlo”



Znakovlje i oznake

- MATEMATIČKA INDUKCIJA
 - tehnika dokazivanja u dva koraka
 - prvo se dokaže **baza** $P(n_0)$
 - zatim se dokaže **induktivna hipoteza** $P(n)$
- RELACIJE
 - binarna relacija R je skup parova, piše se **aRb**
 - članovi para su a iz **domene** i b iz **kodomene**
 - domena i kodomena mogu biti isti skup S

Znakovlje i oznake

- SVOJSTVA RELACIJE NAD S
 - **refleksivnost**: za sve a vrijedi aRa
 - **nerefleksivnost**: ni za jedan a ne vrijedi aRa
 - **tranzitivnost**: za sve a, b, c iz aRb i bRc vrijedi aRc
 - **simetričnost**: ako za sve aRb vrijedi bRa
 - **asimetričnost** ako za sve aRb ne vrijedi bRa
 - asimetrična relacija je ujedno nerefleksivna
 - **ekvivalencija** je refleksivna, simetrična i tranzitivna

1.2. Jezik, automat i gramatika

DEFINICIJI FORMALNOG JEZIKA

- pridružujemo **automat**
- pridružujemo **gramatiku**

AUTOMAT M

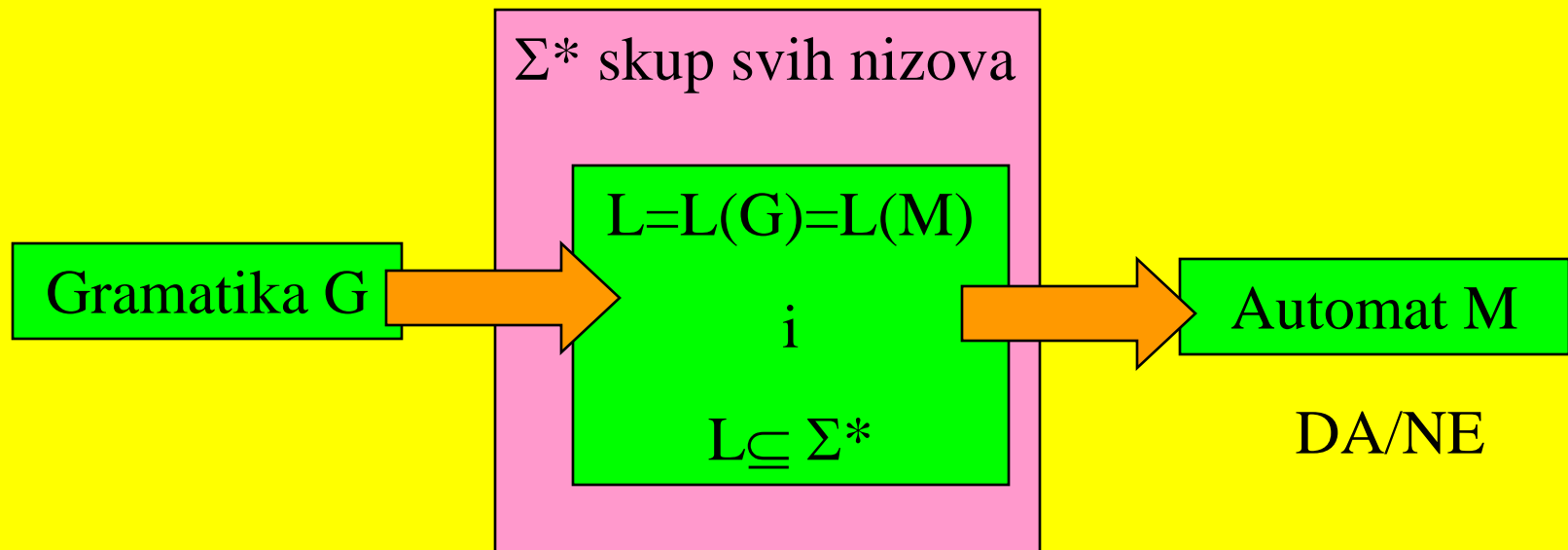
- je matematička struktura koja prihvata jezik $L(M)$
- čita niz znak po znak i odlučuje da li je element jezika
- počinje iz početnog stanja i tijekom rada ih mijenja
- prihvatljiva stanja iskazuju da je niz element
- neprihvatljiva odbacuju niz

Jezik, automat i gramatika

- GRAMATIKA G

- je matematička struktura koja generira jezik $L(G)$
- nizovi se generiraju primjenom pravila **produkcija**
- produkcije se zadaju **završnim** i **nezavršnim** znakovima
- završni (terminalni) znakovi su znakovi abecede jezika
- počevši od početnog nezavršnog znaka
 - primjenom podukcija
 - nastoje se zamijeniti nezavršni znakovi završnim
 - postupak se ponavlja dok se ne dobije niz završnih znakova
- skup svih nizova koje generira gramatika je jezik

Jezik, automat i gramatika



Jezik, automat i gramatika

- L nad abecedom $D=\{0,1,2,3,4,5,6,7,8,9\}$ sadrži brojeve djeljive sa 3 $L=\{\epsilon,0,3,6,9,12,15,\dots\}$

D^* skup svih nizova nad abecedom D

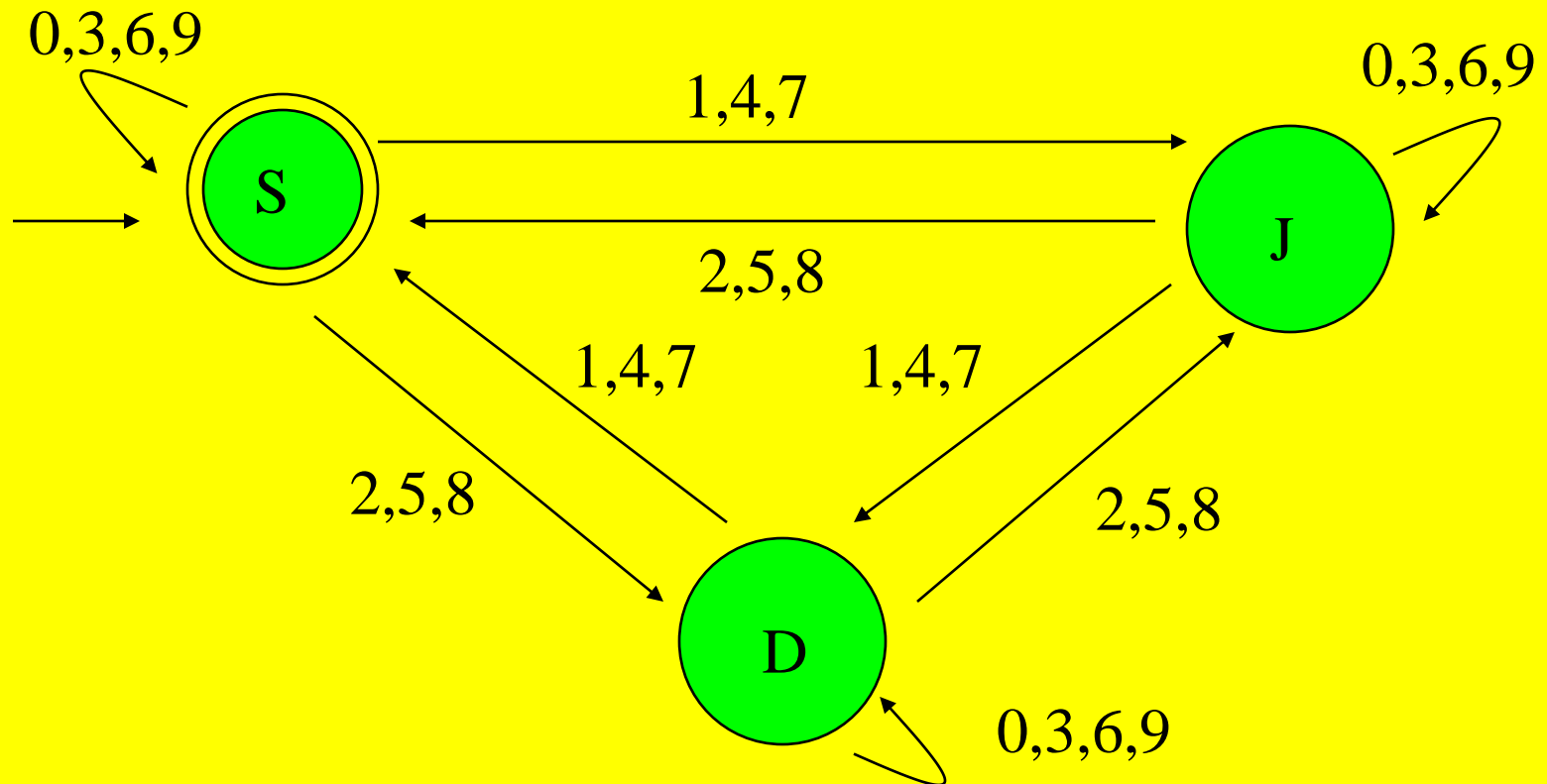
$$D^*=\{\epsilon,1,2,3,4,\dots\}$$

Jezik $L \subseteq D^*$

$$L=\{\epsilon,0,3,6,9,12,15,\dots\}$$

Jezik, automat i gramatika

- Graf automata M za jezik L:



Jezik, automat i gramatika

- Tablica prijelaza automata M za jezik L:

		Ulazni znak										Prihvatljivost
		0	1	2	3	4	5	6	7	8	9	
Stanje	S	S	J	D	S	J	D	S	J	D	S	1
	J	J	D	S	J	D	S	J	D	S	J	0
	D	D	S	J	D	S	J	D	S	J	D	0

Jezik, automat i gramatika

- Gramatika G za jezik L: skup produkcija

$$\langle S \rangle \rightarrow 0\langle S \rangle \mid 3\langle S \rangle \mid 6\langle S \rangle \mid 9\langle S \rangle \mid 1\langle J \rangle \mid 4\langle J \rangle \mid 7\langle J \rangle \mid 2\langle D \rangle \mid 5\langle D \rangle \mid 8\langle D \rangle$$
$$\langle S \rangle \rightarrow \varepsilon$$
$$\langle J \rangle \rightarrow 0\langle J \rangle \mid 3\langle J \rangle \mid 6\langle J \rangle \mid 9\langle J \rangle \mid 1\langle D \rangle \mid 4\langle D \rangle \mid 7\langle D \rangle \mid 2\langle S \rangle \mid 5\langle S \rangle \mid 8\langle S \rangle$$
$$\langle D \rangle \rightarrow 0\langle D \rangle \mid 3\langle D \rangle \mid 6\langle D \rangle \mid 9\langle D \rangle \mid 1\langle S \rangle \mid 4\langle S \rangle \mid 7\langle S \rangle \mid 2\langle J \rangle \mid 5\langle J \rangle \mid 8\langle J \rangle$$

- $\langle \rangle$ su nezavršni znakovi gramatike
- $\langle S \rangle$ je početni nezavršni znak
- \rightarrow znači moguću zamjenu
- \mid je operator “ili”

Jezik, automat i gramatika

- Proces generiranja jezika na osnovu gramatike G

$\langle S \rangle \Rightarrow 1 \langle J \rangle$

$1 \langle J \rangle \Rightarrow 14 \langle D \rangle$

$14 \langle D \rangle \Rightarrow 140 \langle D \rangle$

$140 \langle D \rangle \Rightarrow 1404 \langle S \rangle$

$1404 \langle S \rangle \Rightarrow 1404\varepsilon = 1404$

- \Rightarrow označava proces generiranja niza iz lijevog u desni primjenom točno jedne produkcije
- skraćeno se piše:

$\langle S \rangle \Rightarrow 1 \langle J \rangle \Rightarrow 14 \langle D \rangle \Rightarrow 140 \langle D \rangle \Rightarrow 1404 \langle S \rangle \Rightarrow 1404\varepsilon = 1404$