

Programski prevodioci (vežbe)

Leksička analiza

Zadaci leksičkog analizatora

- Osnovni zadaci leksičkog analizatora:
 - ❑ Izdvajanje reči iz ulaznog koda,
 - ❑ Određivanje značenja izdvojenih reči,
 - ❑ Obrada izdvojenih simbola
(određivanje vrednosti konstanti, upis identifikatora u tabelu simbola,...)
- Sekundarni zadaci leksičkog analizatora:
 - ❑ Uklanjanje belih simbola iz koda (blanko znaka, tabulatora, prelaza na novi red),
 - ❑ Uklanjanje komentara iz koda.

Rezultat leksičkog analizatora

Primer ulaznog koda:

```
if ( a < 3.5 )  
//ako je a ispod granice  
    b_2 = 5;
```

Rezultat leksičke analize:

Reč	Tip tokena	Vrednost
if	IF	-
(LEVA_ZAGRADA	-
a	ID	"a"
<	MANJE	-
3.5	KONSTANTA	3.5 (float)
)	DESNA_ZAGRADA	-
b_2	ID	"b_2"
=	DODELA	-
5	KONSTANTA	5(int)
;	TAČKA_ZAREZ	-

Faze razvoja leksičkog analizatora

- Definisanje konačnog automata za prepoznavanje reči
- Implementacija konačnog automata

Primer: Leksički analizator jezika μ Pascal

$Program \rightarrow \mathbf{program} (NizImena) ;$
 $\quad DeklProm Blok .$

$DeklProm \rightarrow \mathbf{var} NizDekl$

$NizDekl \rightarrow NizDekl Deklaracija$
 $\quad | Deklaracija$

$Deklaracija \rightarrow NizImena : Tip$

$NizImena \rightarrow NizImena , \mathbf{ID} | \mathbf{ID}$

$Tip \rightarrow \mathbf{integer} | \mathbf{char}$

$Blok \rightarrow \mathbf{begin} NizNar \mathbf{end}$

$NizNar \rightarrow NizNar ; Naredba$
 $\quad | Naredba$

$Naredba \rightarrow Ulaz | Izlaz | Dodela$
 $\quad | Blok | IfNar$

$Ulaz \rightarrow \mathbf{read} (\mathbf{ID})$

$Izlaz \rightarrow \mathbf{write} (Izraz)$

$Dodela \rightarrow \mathbf{ID} := Izraz$

$Izraz \rightarrow Izraz + Pizraz | Pizraz$

$Pizraz \rightarrow Pizraz * Fizraz | Fizraz$

$Fizraz \rightarrow \mathbf{ID} | \mathbf{CONST} | (Izraz)$

$IfNar \rightarrow \mathbf{if} Izraz \mathbf{then} Naredba$
 $\quad \mathbf{else} Naredba$

Reči jezika μ Pascal

- Ključne reči:

**program, var, integer, char, begin, end,
read, write, if, then, else**

- Identifikatori:

Niz slova i brojeva u kojem je prvi znak slovo.

- Konstante:

- Tipa integer - Niz dekadnih cifara.
- Tipa char – Znak naveden između apostrofa.

- Operatori:

+, *, :=

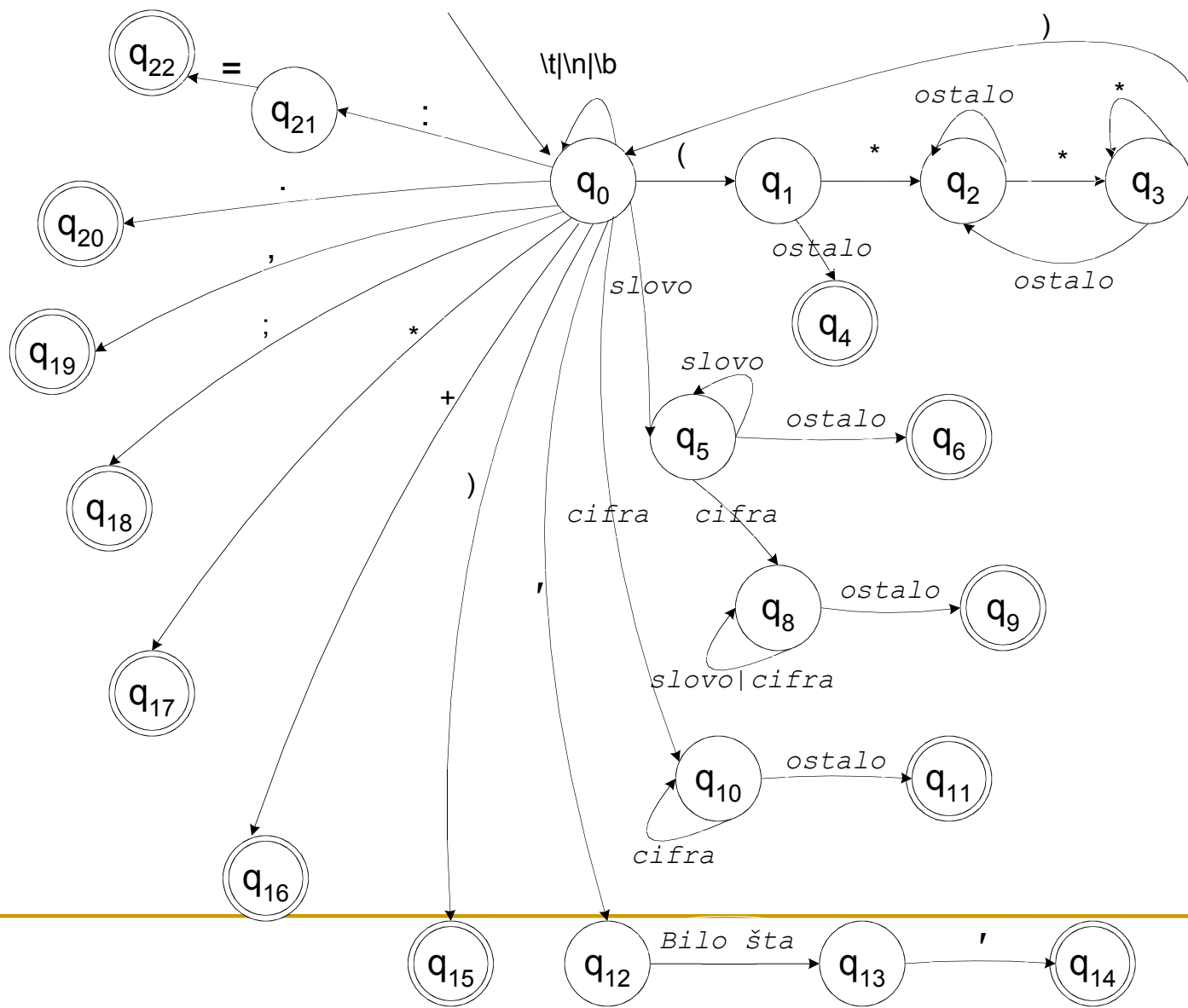
- Separatori:

(,), :, ,, :, .

- Komentari:

Niz karaktera između simbola (i *).*

Graf prelaza konačnog automata



Implementacija leksičkog analizatora

- Na asemblerskom jeziku
- Na nekom višem programskom jeziku za koji prevodilac već postoji
 - Korišćenjem proceduralnih jezika
 - Korišćenjem objektno-orijentisanih jezika
- Korišćenjem alata za automatsko generisanje leksičkih analizatora

Implementacija konačnih automata korišćenjem proceduralnih jezika

```
int DKA(char *ulaz)
{
    enum Stanje { START, ... };
    enum Stanje tabelaPrelaza[brStanja][brSimbola]={ ...};
    int znak;
    enum Stanje tekuce = START;
    while(1)
    {
        znak = getNum(*ulaz++)
        tekuce=tabela[tekuce][znak];
        if ( tekuce==-1 ) return 0;
        if ( znak == EOF )
            break;
    }
    if ( jeZavrsno( tekuce ) )
        return 1;
    return 0;
}
```

Implementacija konačnih automata korišćenjem OO jezika (projektni obrazac State)

