

1. Definirati:

- (a) Pojam binarne relacije na skupu A . Pojam tranzitivnog zatvorenja relacije. Relaciju izvođenja \Rightarrow u gramatici G . Relaciju \Rightarrow^* . Jezik gramatike G .
- (b) Levi količnik $x^{-1}L$, $x \in \Sigma^*$, $L \subseteq \Sigma^*$
- (c) Ručku rečenične forme.
- (d) Skupove $First(A)$ i $Follow(A)$ za neterminal A .

2. Razmatrajmo problem sabiranja binarnih brojeva. Npr.

```
01010 +
01100
-----
10110
```

Postupak ovog sabiranja je moguće opisati trojkama binarnih cifara i to počevši od cifre najmanje težine. Za dati primer su to trojke $(0,0,0)$ $(1,0,1)$ $(0,1,1)$ $(1,1,0)$ $(0,0,1)$. Ukoliko se izostave zagrade i zapete dobija se niska iz $\Sigma = \{0,1\}^*$. Za dati primer se dobija niska 000101011110001. Konstruisati konačni automat koji prepoznaje sve ovakve niske koje predstavljaju ispravna sabiranja. Da li je moguće napraviti ovakav konačni automat ukoliko bi se trojke cifara navodile počevši od cifre najveće težine?

3. Neka je $\Sigma = \{0,1\}$. Za proizvoljnu reč $w \in \Sigma^*$ definišimo $d(w)$ kao nisku koja se dobija od niske w time što se svako 0 zameni sa 00 i svako 1 zameni sa 11. Dalje, neka je $d(L) = \{d(w) \mid w \in L\}$. Ako je L regularan jezik, da li je i $d(L)$ regularan jezik? Dokazati.

4. Data je gramatika G :

$$\begin{aligned} A &\rightarrow BC \\ B &\rightarrow bB \mid \varepsilon \\ C &\rightarrow Cc \mid c \end{aligned}$$

- (a) Konstruisati SLR(1) parser za gramatiku G i odrediti ACTION i GOTO tablice.
- (b) Simulirati rad konstruisanog parsera na prihvatanju niski cc i bbc .
- (c) Na osnovu konstruisanog parsera, napisati program koji proverava da li data niska pripada jeziku $L(G)$.

5. LL(1) gramatikom opisati iterativne **while** iskaze u PASCAL-u i odrediti skupove izbora. Npr.

```
while i < 10 do          while i < n do
  i := i + 1              begin
                           j := i + 1;
                           while j < n do
                             j := j + 1
                           end
```