

Prevodioci i interpretatori - Mart 2004.

1. Regularnim izrazom i minimalnim determinističkim konačnim automatom opisati jezik $L \subset \{0, 1\}^*$ koji se sastoji od reči u kojima se nula pojavljuje paran, a jedinica neparan broj puta.
2. (a) LL(1) gramatikom opisati uslovne dodele u jeziku C, pri čemu je reč o dodeli numeričkoj promenljivoj. Npr.

$a = b > 1 ? 2+c*3 : 4-(c+a+3);$

- (b) Napisati program koji metodom rekursivnog spusta prevodi ispravnu uslovnu dodelu u odgovarajući if-iskaz. Gornji primer treba prevesti u

```
if (b>1)
    a = 2+c*3;
else
    a = 4-(c+a+3);
```

3. Na jeziku C napisati program koji broji redove, reči i znakove unete sa standardnog ulaza, pri čemu se pod rečju podrazumeva bilo koji niz znakova koji ne sadrži razmak, tabulator ili novi red.

Prevodioci i interpretatori - Mart 2004.

1. Regularnim izrazom i minimalnim determinističkim konačnim automatom opisati jezik $L \subset \{0, 1\}^*$ koji se sastoji od reči u kojima se nula pojavljuje paran, a jedinica neparan broj puta.
2. (a) LL(1) gramatikom opisati uslovne dodele u jeziku C, pri čemu je reč o dodeli numeričkoj promenljivoj. Npr.

$a = b > 1 ? 2+c*3 : 4-(c+a+3);$

- (b) Napisati program koji metodom rekursivnog spusta prevodi ispravnu uslovnu dodelu u odgovarajući if-iskaz. Gornji primer treba prevesti u

```
if (b>1)
    a = 2+c*3;
else
    a = 4-(c+a+3);
```

3. Na jeziku C napisati program koji broji redove, reči i znakove unete sa standardnog ulaza, pri čemu se pod rečju podrazumeva bilo koji niz znakova koji ne sadrži razmak, tabulator ili novi red.

Prevodioci i interpretatori - Mart 2004.

1. Regularnim izrazom i minimalnim determinističkim konačnim automatom opisati jezik $L \subset \{0, 1\}^*$ koji se sastoji od reči u kojima se nula pojavljuje paran, a jedinica neparan broj puta.
2. (a) LL(1) gramatikom opisati uslovne dodele u jeziku C, pri čemu je reč o dodeli numeričkoj promenljivoj. Npr.

$a = b > 1 ? 2+c*3 : 4-(c+a+3);$

- (b) Napisati program koji metodom rekursivnog spusta prevodi ispravnu uslovnu dodelu u odgovarajući if-iskaz. Gornji primer treba prevesti u

```
if (b>1)
    a = 2+c*3;
else
    a = 4-(c+a+3);
```

3. Na jeziku C napisati program koji broji redove, reči i znakove unete sa standardnog ulaza, pri čemu se pod rečju podrazumeva bilo koji niz znakova koji ne sadrži razmak, tabulator ili novi red.

4. Za neterminal X gramatike $G = (N, \Sigma, P, S)$, kažemo da je *beskoristan(useless)*, ako ne postoji izvodjenje oblika

$$S \Rightarrow^* wXy \Rightarrow^* wxy$$

gde su $w, y, x \in \Sigma^*$.

Kontekstno slobodna gramatika G je opisana u tekstualnoj datoteci na sledeći način :

- $\Sigma = \{a, b, \dots, z, A, B, \dots, Z\}^+$, tj. terminali su neprazne reči engleske abecede. npr.
Matemacki, fakultet, Beograd
- Neterminali su reči iz Σ , zapisani izmedju znakova $< i >$. npr.
 $<A>$, $$, $<PREVODIOCI>$, $<Zdravo>$.
- Pravila su terminisana sa $;$, a više desnih strana pravila za jedan neterminal je razdvojeno sa $|$. Npr.

```
<IZRAZ> -> <IZRAZ> operator <TERM>
          | <TERM>;
```

Napisati program koji oslobadja gramatiku G beskorisnih neterminala. Koristiti LEX i YACC. Opisati i postupak dobijanja izvršnog koda programa.

4. Za neterminal X gramatike $G = (N, \Sigma, P, S)$, kažemo da je *beskoristan(useless)*, ako ne postoji izvodjenje oblika

$$S \Rightarrow^* wXy \Rightarrow^* wxy$$

gde su $w, y, x \in \Sigma^*$.

Kontekstno slobodna gramatika G je opisana u tekstualnoj datoteci na sledeći način :

- $\Sigma = \{a, b, \dots, z, A, B, \dots, Z\}^+$, tj. terminali su neprazne reči engleske abecede. npr.
Matemacki, fakultet, Beograd
- Neterminali su reči iz Σ , zapisani izmedju znakova $< i >$. npr.
 $<A>$, $$, $<PREVODIOCI>$, $<Zdravo>$.
- Pravila su terminisana sa $;$, a više desnih strana pravila za jedan neterminal je razdvojeno sa $|$. Npr.

```
<IZRAZ> -> <IZRAZ> operator <TERM>
          | <TERM>;
```

Napisati program koji oslobadja gramatiku G beskorisnih neterminala. Koristiti LEX i YACC. Opisati i postupak dobijanja izvršnog koda programa.

4. Za neterminal X gramatike $G = (N, \Sigma, P, S)$, kažemo da je *beskoristan(useless)*, ako ne postoji izvodjenje oblika

$$S \Rightarrow^* wXy \Rightarrow^* wxy$$

gde su $w, y, x \in \Sigma^*$.

Kontekstno slobodna gramatika G je opisana u tekstualnoj datoteci na sledeći način :

- $\Sigma = \{a, b, \dots, z, A, B, \dots, Z\}^+$, tj. terminali su neprazne reči engleske abecede. npr.
Matemacki, fakultet, Beograd
- Neterminali su reči iz Σ , zapisani izmedju znakova $< i >$. npr.
 $<A>$, $$, $<PREVODIOCI>$, $<Zdravo>$.
- Pravila su terminisana sa $;$, a više desnih strana pravila za jedan neterminal je razdvojeno sa $|$. Npr.

```
<IZRAZ> -> <IZRAZ> operator <TERM>
          | <TERM>;
```

Napisati program koji oslobadja gramatiku G beskorisnih neterminala. Koristiti LEX i YACC. Opisati i postupak dobijanja izvršnog koda programa.