

Prevodioci i interpretatori - septembar 2003. - predlog zadataka

1. Data je azbuka $\Sigma = \{0, 1\}$. Odrediti sve reči $x \in \Sigma^*$ za koje važi

$$0x = x0$$

2. Konačnim automatom i regularnim izrazom opisati skup reči iz $\{0, 1\}^*$ takve da im 101 nije podreč. Ukoliko konstruisani automat nije minimalni deterministički, minimizovati ga i determinizovati.
3. Dokazati ili opovrgnuti:
- Klasa regularnih jezika je zatvorena u odnosu na operaciju simetrične razlike.
 - Klasa kontekst slobodnih jezika je zatvorena u odnosu na operaciju preseka.
4. Napisati program koristeći Lex koji za uneti rimski broj proverava da li je ispravan i ako jeste ispisuje njegovu vrednost.
5. Sastaviti LL(1) gramatiku za jezik aritmetičkih izraza nad azbukom $\{ (,), [,], +, -, *, /, 'a, b \}$ pri čemu se običe koriste za zagradjivanje izraza bez zagrada, dok se uglaste zagrada koriste za zagradjivanje izraza koji već u sebi sadrže neke zagrade. Za ovako napravljenu gramatiku odrediti skupove izbora i na osnovu njih u programskom jeziku C skicirati implementaciju potisnog automata koji za uneti izraz ispituje da li pripada jeziku ove gramatike.
6. Gramatiku

$$\begin{aligned} S &\rightarrow Ba|Ab \\ A &\rightarrow Sa|AAb|a \\ B &\rightarrow Sb|BBa|b \end{aligned}$$

prevesti u Greibach normalnu formu

7. Napisati program koji aritmetički izraz koji se unosi sa ulaza prevodi u tzv. MathML notaciju, na sledeći način. Oznake svih promenljivih (u obzir uzeti samo jednoslovne promenjive) koje se javljaju u izrazu se pišu unutar `<mi>` etiketa. Oznake svih brojeva se pišu unutar `<mn>` etiketa. Oznake aritmetičkih operacija `+`, `-` i `*` se pišu unutar `<mo>` etiketa. Svaki izraz koji se pojavi u zagradama se stavlja unutar `<mfenced>` etiketa. Ukoliko se naidje na operaciju deljenja ona se obradjuje tako što se unutar etiketa `<mfrac>` navedu deljenik i delilac. Svi izrazi koji se prilikom uobičajenog načina ispisa pisali u jednom redu se stavljaju unutar `<mrow>` etiketa (to su izrazi koji sadrže niz operacija sabiranja, oduzimanja i množenja). Npr. izraz $(x+y/5-4)/20+2$ je potrebno prevesti u

```
<mrow>
  <mfrac>
    <mfenced>
      <mrow>
        <mi>x</mi>
        <mo>+</mo>
        <mfrac>
          <mi>y</mi>
          <mn>5</mn>
        </mfrac>
        <mo>-</mo>
        <mn>4</mn>
      </mrow>
    </mfenced>
    <mn>20</mn>
  </mfrac>
  <mo>+</mo>
  <mn>2</mn>
</mrow>
```

Pokušajte da ispoštujete način nazublivanja ispisa koji je dat u primeru.

8. Pretpostavimo da `main()` funkcija programa napisana na jeziku C++ koristi klasu `String`. Funkcija počinje nizom deklaracija promenljivih za kojim sledi niz izraza nad tim promenjivim. Klasa `String` ima konstruktor na osnovu jednog argumenta tipa `char*`, destruktor, zatim operatore `+`, i `+=` za konkatenciju i funkciju `size()` koja vraća veličinu stringa. Napraviti program koji će da prepíše taj `main` tako da se umesto klase `String` radi samo sa niskama karaktera i pokazivačima na njih. Npr.

```

main()
{   int a;
    float b;
    String c("Niska"),d;
    String e;

    c+="Cao";

    a++;

    d=c;

    e=d+"Ahoj";

}

```

```

main()
{   int a;
    float b;
    char *c="Niska", *d=NULL;
    char *e=NULL;

    char *tmp;
    tmp=new char[strlen(c)+3+1];
    strcpy(tmp,c);
    strcpy(tmp+strlen(c),"Cao");
    if (c) delete [] c;
    c=tmp;

    a++;

    tmp=new char[strlen(c)+1];
    strcpy(tmp,c);
    if (d) delete[] d;
    d=tmp;

    tmp=new char[strlen(d)+4+1];
    strcpy(tmp,d);
    strcpy(tmp+strlen(d),"Ahoj");
    if (e) delete[] e;
    e=tmp;

    if (c) delete []c;
    if (d) delete []d;
    if (e) delete []e;
}

```

Način prevodjenja odredite sami i pokušajte da pri tome vodite računa o optimalnosti.