

Prevodioci i interpretatori - mart 2005.
teorijski deo

1. Minimalnim determinističkim konačnim automatom i regularnim izrazom opisati jezik svih binarnih brojeva koji imaju svojstvo da su deljivi sa tri, a da ne sadrže podreč 101.
2. Opisati LL(1) gramatikom iterativne iskaze jezika C. Telo iterativnih iskaza ograničiti tako da mogu da sadrže samo druge iterativne iskaze ili iskaze dodele aritmetičkih izraza promenljivim. Uslove ograničiti samo na osnove relacije (<, >, ==, !=, ...) nad aritmetičkim izrazima. Npr.

```
for (i = 0; i < n-1; i = i+1)
{
    j = i+1;
    while(j < n)
        j = j+1;
}
```

3. Kontekstno slobodnom gramatikom opisati ispravne opise HTML tabela. Na osnovu gramatike, konstruisati postisni automat koji metodom sintaksne analize navise prihvata pomenuti jezik. Pretpostaviti da se između etiketa nalazi tekst prepoznat u fazi leksičke analize kao token `cdata`. Simulirati rad potisnog automata i napisati najdešnje izvodenje za tabelu:

```
<table>
  <tr><td>cdata</td><td>cdata</td></tr>
  <tr><td>cdata</td></tr>
</table>
```

Da li je pomenuti jezik regularan? Ukoliko jeste, opisati ga regularnim izrazom, a ukoliko nije, dokazati da nije.

Prevodioci i interpretatori - mart 2005.
teorijski deo

1. Minimalnim determinističkim konačnim automatom i regularnim izrazom opisati jezik svih binarnih brojeva koji imaju svojstvo da su deljivi sa tri, a da ne sadrže podreč 101.
2. Opisati LL(1) gramatikom iterativne iskaze jezika C. Telo iterativnih iskaza ograničiti tako da mogu da sadrže samo druge iterativne iskaze ili iskaze dodele aritmetičkih izraza promenljivim. Uslove ograničiti samo na osnove relacije (<, >, ==, !=, ...) nad aritmetičkim izrazima. Npr.

```
for (i = 0; i < n-1; i = i+1)
{
    j = i+1;
    while(j < n)
        j = j+1;
}
```

3. Kontekstno slobodnom gramatikom opisati ispravne opise HTML tabela. Na osnovu gramatike, konstruisati postisni automat koji metodom sintaksne analize navise prihvata pomenuti jezik. Pretpostaviti da se između etiketa nalazi tekst prepoznat u fazi leksičke analize kao token `cdata`. Simulirati rad potisnog automata i napisati najdešnje izvodenje za tabelu:

```
<table>
  <tr><td>cdata</td><td>cdata</td></tr>
  <tr><td>cdata</td></tr>
</table>
```

Da li je pomenuti jezik regularan? Ukoliko jeste, opisati ga regularnim izrazom, a ukoliko nije, dokazati da nije.

Prevodioci i interpretatori - mart 2005.
teorijski deo

1. Minimalnim determinističkim konačnim automatom i regularnim izrazom opisati jezik svih binarnih brojeva koji imaju svojstvo da su deljivi sa tri, a da ne sadrže podreč 101.
2. Opisati LL(1) gramatikom iterativne iskaze jezika C. Telo iterativnih iskaza ograničiti tako da mogu da sadrže samo druge iterativne iskaze ili iskaze dodele aritmetičkih izraza promenljivim. Uslove ograničiti samo na osnove relacije (<, >, ==, !=, ...) nad aritmetičkim izrazima. Npr.

```
for (i = 0; i < n-1; i = i+1)
{
    j = i+1;
    while(j < n)
        j = j+1;
}
```

3. Kontekstno slobodnom gramatikom opisati ispravne opise HTML tabela. Na osnovu gramatike, konstruisati postisni automat koji metodom sintaksne analize navise prihvata pomenuti jezik. Pretpostaviti da se između etiketa nalazi tekst prepoznat u fazi leksičke analize kao token `cdata`. Simulirati rad potisnog automata i napisati najdešnje izvodenje za tabelu:

```
<table>
  <tr><td>cdata</td><td>cdata</td></tr>
  <tr><td>cdata</td></tr>
</table>
```

Da li je pomenuti jezik regularan? Ukoliko jeste, opisati ga regularnim izrazom, a ukoliko nije, dokazati da nije.

Prevodioci i interpretatori - mart 2005.
praktični deo

1. Napisati perl skript koji za sve HTML datoteke koje se nalaze u direktorijumu čije se ime navodi kao argument komandne linije generiše odgovarajuće "obojene" HTML datoteke tj. HTML datoteke koje služe za pregledno prikazivanje HTML koda na webu. Potrebno je da sve etikete budu podebljane i obojene jednom bojom, imena njihovih atributa drugom, a vrednosti atributa trećom bojom. Pri tom, svi brojevi koji se javljaju u okviru etiketa bi trebalo da budu iskošeni.
2. Napraviti interpreter za minijturni skript jezik koji pomalo liči na jezik PERL. Jezik odlikuje skalarni tip podataka koji obuhvata cele brojeve i niske karaktera, i nizovni tip koji predstavlja liste skalara. Promenljive u programu se ne deklariraju, ali važi ograničenje da imena skalarnih promenljivih počinju znakom \$ (npr. \$scalar), dok sve liste počinju simbolom @ (npr. @list). Jedina naredba je naredba dodele.

Nad brojevnim skalarima su definisane uobičajene aritmetičke operacije, dok su nad niskama definisane operacije konkatencije ., množenja brojem x koje označava konkatenciju određenog broja puta, kao i operator izdvajanja podnisk ([m,n]) koji izdvaja n karaktera počevši od pozicije m . Brojevni skalari se implicitno konvertuju u niske karaktera koje sadrže njihovu tekstualnu reprezentaciju, dok se niske konvertuju u cele brojeve koje opisuju (odnosno 0 ukoliko ne sadrže opis broja).

Nad listama su definisane funkcije **push** koja ubacuje dati skalar na kraj date liste i **pop** koja vraća poslednji element liste. Nad listama je definisan i operator indeksnog pristupa [n] koji vraća n -ti element liste.

Jezik poseduje funkciju **print** koja služi za ispis.

Npr.

```
$broj = 2;           # $broj postaje 2
$string = "abc";     # $string postaje "abc"
@list = ($broj, "def"); # @list postaje (2, "def")
$string = ($string x ($broj+1))."ab"[0,1]; # $string postaje "abcabcabca"
push(@list, $string); # @list postaje (2, "def", "abcabcabca")
print(@list);        # ispisuje se (2, "def", "abcabcabca")
```

Prevodioci i interpretatori - mart 2005.
praktični deo

1. Napisati perl skript koji za sve HTML datoteke koje se nalaze u direktorijumu čije se ime navodi kao argument komandne linije generiše odgovarajuće "obojene" HTML datoteke tj. HTML datoteke koje služe za pregledno prikazivanje HTML koda na webu. Potrebno je da sve etikete budu podebljane i obojene jednom bojom, imena njihovih atributa drugom, a vrednosti atributa trećom bojom. Pri tom, svi brojevi koji se javljaju u okviru etiketa bi trebalo da budu iskošeni.
2. Napraviti interpreter za minijturni skript jezik koji pomalo liči na jezik PERL. Jezik odlikuje skalarni tip podataka koji obuhvata cele brojeve i niske karaktera, i nizovni tip koji predstavlja liste skalara. Promenljive u programu se ne deklariraju, ali važi ograničenje da imena skalarnih promenljivih počinju znakom \$ (npr. \$scalar), dok sve liste počinju simbolom @ (npr. @list). Jedina naredba je naredba dodele.

Nad brojevnim skalarima su definisane uobičajene aritmetičke operacije, dok su nad niskama definisane operacije konkatencije ., množenja brojem x koje označava konkatenciju određenog broja puta, kao i operator izdvajanja podnisk ([m,n]) koji izdvaja n karaktera počevši od pozicije m . Brojevni skalari se implicitno konvertuju u niske karaktera koje sadrže njihovu tekstualnu reprezentaciju, dok se niske konvertuju u cele brojeve koje opisuju (odnosno 0 ukoliko ne sadrže opis broja).

Nad listama su definisane funkcije **push** koja ubacuje dati skalar na kraj date liste i **pop** koja vraća poslednji element liste. Nad listama je definisan i operator indeksnog pristupa [n] koji vraća n -ti element liste.

Jezik poseduje funkciju **print** koja služi za ispis.

Npr.

```
$broj = 2;           # $broj postaje 2
$string = "abc";     # $string postaje "abc"
@list = ($broj, "def"); # @list postaje (2, "def")
$string = ($string x ($broj+1))."ab"[0,1]; # $string postaje "abcabcabca"
push(@list, $string); # @list postaje (2, "def", "abcabcabca")
print(@list);        # ispisuje se (2, "def", "abcabcabca")
```