

### Prevodioci i interpretatori - januar 2006. - teorijski deo ispita

1. Ispitati da li su jezici  $L_1 = (a^2 + b^*a^+)^*$  i  $L_2 = ((a^2)^* + b^*a)^*$  jednaki. Odgovor dokazati.
2. LL(1) gramatikom opisati niz definicija listi i heš tabela u jeziku PERL. Npr.

```
@1 = ("Pii", 45, 38.3);  
%a = {"Pera" => 25, "Mika" => 30, "Ana" => 40};
```

Za konstruisanu gramatiku odrediti skupove izbora i konstruisati potisni automat koji tehnikom analize naniže prihvata dati jezik. Za nisku %a = {"Pera" => 25, "Ana" => 40}; odrediti najlevlje izvođenje, nacrtati drvo izvođenja i prikazati rad automata prilikom njenog prihvatanja.

3. KS gramatikom opisati aritmetičke izraze sa operacijama + i \* u kojima se ne koriste zagrade. Kreirati SLR parser za datu gramatiku, opisati ga *action* i *goto* tablicama i prikazati njegov rad na prihvatanju niske 2+3\*5.

### Prevodioci i interpretatori - januar 2006. - teorijski deo ispita

1. Ispitati da li su jezici  $L_1 = (a^2 + b^*a^+)^*$  i  $L_2 = ((a^2)^* + b^*a)^*$  jednaki. Odgovor dokazati.
2. LL(1) gramatikom opisati niz definicija listi i heš tabela u jeziku PERL. Npr.

```
@1 = ("Pii", 45, 38.3);  
%a = {"Pera" => 25, "Mika" => 30, "Ana" => 40};
```

Za konstruisanu gramatiku odrediti skupove izbora i konstruisati potisni automat koji tehnikom analize naniže prihvata dati jezik. Za nisku %a = {"Pera" => 25, "Ana" => 40}; odrediti najlevlje izvođenje, nacrtati drvo izvođenja i prikazati rad automata prilikom njenog prihvatanja.

3. KS gramatikom opisati aritmetičke izraze sa operacijama + i \* u kojima se ne koriste zagrade. Kreirati SLR parser za datu gramatiku, opisati ga *action* i *goto* tablicama i prikazati njegov rad na prihvatanju niske 2+3\*5.

### Prevodioci i interpretatori - januar 2006. - teorijski deo ispita

1. Ispitati da li su jezici  $L_1 = (a^2 + b^*a^+)^*$  i  $L_2 = ((a^2)^* + b^*a)^*$  jednaki. Odgovor dokazati.
2. LL(1) gramatikom opisati niz definicija listi i heš tabela u jeziku PERL. Npr.

```
@1 = ("Pii", 45, 38.3);  
%a = {"Pera" => 25, "Mika" => 30, "Ana" => 40};
```

Za konstruisanu gramatiku odrediti skupove izbora i konstruisati potisni automat koji tehnikom analize naniže prihvata dati jezik. Za nisku %a = {"Pera" => 25, "Ana" => 40}; odrediti najlevlje izvođenje, nacrtati drvo izvođenja i prikazati rad automata prilikom njenog prihvatanja.

3. KS gramatikom opisati aritmetičke izraze sa operacijama + i \* u kojima se ne koriste zagrade. Kreirati SLR parser za datu gramatiku, opisati ga *action* i *goto* tablicama i prikazati njegov rad na prihvatanju niske 2+3\*5.

### Prevodioci i interpretatori - januar 2006. - teorijski deo ispita

1. Ispitati da li su jezici  $L_1 = (a^2 + b^*a^+)^*$  i  $L_2 = ((a^2)^* + b^*a)^*$  jednaki. Odgovor dokazati.
2. LL(1) gramatikom opisati niz definicija listi i heš tabela u jeziku PERL. Npr.

```
@1 = ("Pii", 45, 38.3);  
%a = {"Pera" => 25, "Mika" => 30, "Ana" => 40};
```

Za konstruisanu gramatiku odrediti skupove izbora i konstruisati potisni automat koji tehnikom analize naniže prihvata dati jezik. Za nisku %a = {"Pera" => 25, "Ana" => 40}; odrediti najlevlje izvođenje, nacrtati drvo izvođenja i prikazati rad automata prilikom njenog prihvatanja.

3. KS gramatikom opisati aritmetičke izraze sa operacijama + i \* u kojima se ne koriste zagrade. Kreirati SLR parser za datu gramatiku, opisati ga *action* i *goto* tablicama i prikazati njegov rad na prihvatanju niske 2+3\*5.

4. Data je KS gramatika

$$\begin{array}{lcl} S & \rightarrow & \text{if } c \text{ then } S \text{ else } S \\ & | & \text{while } c \text{ do } S \\ & | & s \end{array}$$

- (a) Ispitati da li je data gramatika višeznačna. Dokazati odgovor.
- (b) Ukoliko je gramatika višeznačna navesti bar jednu njoj ekvivalentnu nevišeznačnu gramatiku.
- (c) Šta možete reći o opštem algoritmu za ispitivanje da li je proizvoljna gramatika višeznačna.
- (d) Dokazati da nijedna LL(1) gramatika ne može biti višeznačna.

4. Data je KS gramatika

$$\begin{array}{lcl} S & \rightarrow & \text{if } c \text{ then } S \text{ else } S \\ & | & \text{while } c \text{ do } S \\ & | & s \end{array}$$

- (a) Ispitati da li je data gramatika višeznačna. Dokazati odgovor.
- (b) Ukoliko je gramatika višeznačna navesti bar jednu njoj ekvivalentnu nevišeznačnu gramatiku.
- (c) Šta možete reći o opštem algoritmu za ispitivanje da li je proizvoljna gramatika višeznačna.
- (d) Dokazati da nijedna LL(1) gramatika ne može biti višeznačna.

4. Data je KS gramatika

$$\begin{array}{lcl} S & \rightarrow & \text{if } c \text{ then } S \text{ else } S \\ & | & \text{while } c \text{ do } S \\ & | & s \end{array}$$

- (a) Ispitati da li je data gramatika višeznačna. Dokazati odgovor.
- (b) Ukoliko je gramatika višeznačna navesti bar jednu njoj ekvivalentnu nevišeznačnu gramatiku.
- (c) Šta možete reći o opštem algoritmu za ispitivanje da li je proizvoljna gramatika višeznačna.
- (d) Dokazati da nijedna LL(1) gramatika ne može biti višeznačna.

4. Data je KS gramatika

$$\begin{array}{lcl} S & \rightarrow & \text{if } c \text{ then } S \text{ else } S \\ & | & \text{while } c \text{ do } S \\ & | & s \end{array}$$

- (a) Ispitati da li je data gramatika višeznačna. Dokazati odgovor.
- (b) Ukoliko je gramatika višeznačna navesti bar jednu njoj ekvivalentnu nevišeznačnu gramatiku.
- (c) Šta možete reći o opštem algoritmu za ispitivanje da li je proizvoljna gramatika višeznačna.
- (d) Dokazati da nijedna LL(1) gramatika ne može biti višeznačna.

## Prevodioci i interpretatori - januar 2006. - praktični deo ispita

1. Korišćenjem jezika PERL napisati konverter ispravno zapisanih HTML datoteka u pregledan, tekstualni format. Konverter treba da obradi sledeće etikete:

- (a) `<img>` se zamenjuje imenom slike između vitičastih zagrada. Npr. `` se zamenjuje sa `{slika.jpg}`
- (b) `<a>` linkovi se zamenjuju sa adresom na koju ukazuju između uglastih zagrada Npr. `<a href="http://www.google.com">` se zamenjuje sa `[http://www.google.com]`
- (c) `<h1>` naslovi se ispisuju se sa po jednim razmakom između svakog slova i po jednim praznim redom ispred i iza (npr. naslov `<h1>OV0 JE NASLOV</h2>` se prevodi u `O V O J E N A S L O V`)
- (d) `<h2>` do `<h6>` naslovi se ispisuju sa po jednim praznim redom ispred i iza.
- (e) `<p>` se zamenjuje jednim praznim redom
- (f) `<br>` se zamenjuje prelaskom u novi red
- (g) `<ol>` nabrojanja se zamenjuju brojevima. Npr.

```
<ol><li>Prva stavka<li>Druga<li>Trec</ol>
```

se prikazuje kao

- 1. Prva stavka
- 2. Druga
- 3. Trec

Liste mogu biti ugnježdene.

Sve višestruke beline se zamenjuju jednostrukim, prelasci u novi red se ignorišu, a tekst se formatira tako da svaki red zauzima najviše 80 karaktera. [20]

## Prevodioci i interpretatori - januar 2006. - praktični deo ispita

1. Korišćenjem jezika PERL napisati konverter ispravno zapisanih HTML datoteka u pregledan, tekstualni format. Konverter treba da obradi sledeće etikete:

- (a) `<img>` se zamenjuje imenom slike između vitičastih zagrada. Npr. `` se zamenjuje sa `{slika.jpg}`
- (b) `<a>` linkovi se zamenjuju sa adresom na koju ukazuju između uglastih zagrada Npr. `<a href="http://www.google.com">` se zamenjuje sa `[http://www.google.com]`
- (c) `<h1>` naslovi se ispisuju se sa po jednim razmakom između svakog slova i po jednim praznim redom ispred i iza (npr. naslov `<h1>OV0 JE NASLOV</h2>` se prevodi u `O V O J E N A S L O V`)
- (d) `<h2>` do `<h6>` naslovi se ispisuju sa po jednim praznim redom ispred i iza.
- (e) `<p>` se zamenjuje jednim praznim redom
- (f) `<br>` se zamenjuje prelaskom u novi red
- (g) `<ol>` nabrojanja se zamenjuju brojevima. Npr.

```
<ol><li>Prva stavka<li>Druga<li>Trec</ol>
```

se prikazuje kao

- 1. Prva stavka
- 2. Druga
- 3. Trec

Liste mogu biti ugnježdene.

Sve višestruke beline se zamenjuju jednostrukim, prelasci u novi red se ignorišu, a tekst se formatira tako da svaki red zauzima najviše 80 karaktera. [20]

2. Datoteka sadrži niz regularnih definicija. Svaka regularna definicija je oblika `id = regex`, gde u izgradnji regularnog izraza mogu učestvovati karakteri, klase karaktera, prethodno definisani regularni izrazi i operacije `|`, `*`, `+`, `?`. Prilikom izgradnje regularnog izraza moguće je koristiti operator `basic_regex` kojom se dati regularni izraz prevodi u osnovni, odnosno oslobađa karakterskih klasa i operatora `+` i `?`. Funkcijom `print` vrši se ispisivanje datog regularnog izraza. Npr.

```
DIGIT = [0-9];
NUM = {DIGIT}?.{DIGIT};
print({NUM});                [0-9]?.[0-9]
HEX = 0x({DIGIT}|[A-F])+;
DIGIT = [0-1];
print({HEX});                0x([0-9]|[A-F])+
BIN = basic_regex({DIGIT}+);
print({BIN});                (0|1)(0|1)*
```

- (a) Kreirati leksički analizator i gramatikom opisati opisani oblik datoteka. [5]
- (b) Napisati hijerarhiju klasa (C++) ili strukturu (C) za predstavljanje regularnih izraza u obliku stabla. [12]
- (c) Definisati i implementirati funkcije za ispis regularnog izraza. [3]
- (d) Definisati tablicu simbola u kojoj se čuvaju regularne definicije. [3]
- (e) Definisati funkcije za prevođenje regularnih izraza u osnovne. [7]

2. Datoteka sadrži niz regularnih definicija. Svaka regularna definicija je oblika `id = regex`, gde u izgradnji regularnog izraza mogu učestvovati karakteri, klase karaktera, prethodno definisani regularni izrazi i operacije `|`, `*`, `+`, `?`. Prilikom izgradnje regularnog izraza moguće je koristiti operator `basic_regex` kojom se dati regularni izraz prevodi u osnovni, odnosno oslobađa karakterskih klasa i operatora `+` i `?`. Funkcijom `print` vrši se ispisivanje datog regularnog izraza. Npr.

```
DIGIT = [0-9];
NUM = {DIGIT}?.{DIGIT};
print({NUM});                [0-9]?.[0-9]
HEX = 0x({DIGIT}|[A-F])+;
DIGIT = [0-1];
print({HEX});                0x([0-9]|[A-F])+
BIN = basic_regex({DIGIT}+);
print({BIN});                (0|1)(0|1)*
```

- (a) Kreirati leksički analizator i gramatikom opisati opisani oblik datoteka. [5]
- (b) Napisati hijerarhiju klasa (C++) ili strukturu (C) za predstavljanje regularnih izraza u obliku stabla. [12]
- (c) Definisati i implementirati funkcije za ispis regularnog izraza. [3]
- (d) Definisati tablicu simbola u kojoj se čuvaju regularne definicije. [3]
- (e) Definisati funkcije za prevođenje regularnih izraza u osnovne. [7]