

Prevodioci i interpretatori - februar 2006. - teorijski deo ispita

1. Konstruisati konačni transduktor koji određuje celobrojni količnik pri deljenju broja zapisanog u osnovi 6 brojem 5. Npr. za ulaz 150342 transduktor na izlazu daje broj 21154 [13]
2. LL(1) gramatikom opisati deklaracije (i moguću inicijalizacije) jednodimenzionalnih odnosno višedimenzionalnih nizova u jeziku C. Npr.

```
int a[10], b[] = {1, 5, 8};
char s[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

Za konstruisanu gramatiku odrediti skupove izbora i konstruisati potisni automat koji tehnikom analize naniže prihvata dati jezik. Za nisku `int a[10], b[] = {1, 5, 8};` odrediti najlevlje izvođenje, nacrtati drvo izvođenja i prikazati rad automata prilikom njenog prihvatanja. [14]

3. LR gramatikom opisati osnovne regularne izraze sa operacijama konkatenacije, $*$ i $|$ nad azbukom $\Sigma = \{a, b\}$. Kreirati SLR parser za datu gramatiku, opisati ga *action* i *goto* tablicama i prikazati njegov rad na prihvatanju niske $(a|b)*c$. Da li je konstruisana gramatika SLR? [13]
4. Data je KS gramatika $G = (\{S\}, \{a, b\}, P, S)$, gde je P skup $S \rightarrow aSb \mid aaSb \mid \varepsilon$
 - (a) Dokazati da je data gramatika višeznačna.
 - (b) Konstruisati nevišeznačnu gramatiku G' ekvivalentnu datoju.
 - (c) Dokazati da G' nije višeznačna i da je $L(G) = L(G')$ [10]

Prevodioci i interpretatori - januar 2006. - teorijski deo ispita

1. Konstruisati konačni transduktor koji određuje celobrojni količnik pri deljenju broja zapisanog u osnovi 6 brojem 5. Npr. za ulaz 150342 transduktor na izlazu daje broj 21154 [13]
2. LL(1) gramatikom opisati deklaracije (i moguću inicijalizacije) jednodimenzionalnih odnosno višedimenzionalnih nizova u jeziku C. Npr.

```
int a[10], b[] = {1, 5, 8};
char s[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

Za konstruisanu gramatiku odrediti skupove izbora i konstruisati potisni automat koji tehnikom analize naniže prihvata dati jezik. Za nisku `int a[10], b[] = {1, 5, 8};` odrediti najlevlje izvođenje, nacrtati drvo izvođenja i prikazati rad automata prilikom njenog prihvatanja. [14]

3. LR gramatikom opisati osnovne regularne izraze sa operacijama konkatenacije, $*$ i $|$ nad azbukom $\Sigma = \{a, b\}$. Kreirati SLR parser za datu gramatiku, opisati ga *action* i *goto* tablicama i prikazati njegov rad na prihvatanju niske $(a|b)*c$. Da li je konstruisana gramatika SLR? [13]
4. Data je KS gramatika $G = (\{S\}, \{a, b\}, P, S)$, gde je P skup $S \rightarrow aSb \mid aaSb \mid \varepsilon$
 - (a) Dokazati da je data gramatika višeznačna.
 - (b) Konstruisati nevišeznačnu gramatiku G' ekvivalentnu datoju.
 - (c) Dokazati da G' nije višeznačna i da je $L(G) = L(G')$ [10]

Prevodioci i interpretatori - januar 2006. - teorijski deo ispita

1. Konstruisati konačni transduktor koji određuje celobrojni količnik pri deljenju broja zapisanog u osnovi 6 brojem 5. Npr. za ulaz 150342 transduktor na izlazu daje broj 21154 [13]
2. LL(1) gramatikom opisati deklaracije (i moguću inicijalizacije) jednodimenzionalnih odnosno višedimenzionalnih nizova u jeziku C. Npr.

```
int a[10], b[] = {1, 5, 8};
char s[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

Za konstruisanu gramatiku odrediti skupove izbora i konstruisati potisni automat koji tehnikom analize naniže prihvata dati jezik. Za nisku `int a[10], b[] = {1, 5, 8};` odrediti najlevlje izvođenje, nacrtati drvo izvođenja i prikazati rad automata prilikom njenog prihvatanja. [14]

3. LR gramatikom opisati osnovne regularne izraze sa operacijama konkatenacije, $*$ i $|$ nad azbukom $\Sigma = \{a, b\}$. Kreirati SLR parser za datu gramatiku, opisati ga *action* i *goto* tablicama i prikazati njegov rad na prihvatanju niske $(a|b)*c$. Da li je konstruisana gramatika SLR? [13]
4. Data je KS gramatika $G = (\{S\}, \{a, b\}, P, S)$, gde je P skup $S \rightarrow aSb \mid aaSb \mid \varepsilon$
 - (a) Dokazati da je data gramatika višeznačna.
 - (b) Konstruisati nevišeznačnu gramatiku G' ekvivalentnu datoju.
 - (c) Dokazati da G' nije višeznačna i da je $L(G) = L(G')$ [10]

Prevodioci i interpretatori - februar 2006. - praktični deo ispita

1. Sa standardnog ulaza se unosi niz linija od kojih svaka sadrži ispravan aritmetički izraz sa operacijama sabiranja i množenja celih brojeva. Napisati PERL skript koji tehnikom rekursivnog spusta izračunava i ispisuje vrednost svakog unetog izraza. Npr. [15]

```
> (3+6)*2
(3+6)*2 = 18
> 2*(3+1+2)*32
2*(3+1+2)*32 = 384
```

2. Napraviti interpreter za mali jezik iskaznog računa. Jezik dozvoljava definisanje iskaznih formula i iskaznih promenljivih. Iskazne formule se grade od 26 iskaznih slova ([a-z]), konstanti T i F i prethodno definisanih iskaznih promenljivih koje se navode u obliku {prom}. Dozvoljene operacije, u opadajućem prioritetu, su negacija (~), konjunkcija (/\), disjunkcija (\/) i implikacija (=>). Funkcija print(f) ispisuje formulu f na standardni izlaz, substitute(f, p, g) zamenjuje sva pojavljivanja iskaznog slova p u formuli f formulom g, a funkcija value(f, valuation) ispisuje vrednost formule pri datoj valuaciji promenljivih. Valuacije se navode u obliku [p1->v1, ..., pn->vn]. Prijaviti grešku ukoliko navedena valuacija nije dovoljna za određivanje vrednosti formule (Npr. value(p/\q, [p->T])).

```
f1 := (p => (q /\ ~p)) => q;
f2 := {f1} /\ ~q;
f3 := substitute(f2, p, T);
print(f3);                                ((T => (q /\ (~T))) => q) /\ (~q)
value(f1, [p->T, q->F]);                  T
```

- (a) Gramatikom opisati dati jezik i konstruisati leksički analizator. [4]
(b) Definirati hijerarhiju klasa (struktura) za predstavljanje iskaznih formula u obliku stabla. [12]
(c) Implementirati funkciju print. [3]
(d) Implementirati funkciju substitute. [6]
(e) Implementirati funkciju value. [10]

Prevodioci i interpretatori - februar 2006. - praktični deo ispita

1. Sa standardnog ulaza se unosi niz linija od kojih svaka sadrži ispravan aritmetički izraz sa operacijama sabiranja i množenja celih brojeva. Napisati PERL skript koji tehnikom rekursivnog spusta izračunava i ispisuje vrednost svakog unetog izraza. Npr. [15]

```
> (3+6)*2
(3+6)*2 = 18
> 2*(3+1+2)*32
2*(3+1+2)*32 = 384
```

2. Napraviti interpreter za mali jezik iskaznog računa. Jezik dozvoljava definisanje iskaznih formula i iskaznih promenljivih. Iskazne formule se grade od 26 iskaznih slova ([a-z]), konstanti T i F i prethodno definisanih iskaznih promenljivih koje se navode u obliku {prom}. Dozvoljene operacije, u opadajućem prioritetu, su negacija (~), konjunkcija (/\), disjunkcija (\/) i implikacija (=>). Funkcija print(f) ispisuje formulu f na standardni izlaz, substitute(f, p, g) zamenjuje sva pojavljivanja iskaznog slova p u formuli f formulom g, a funkcija value(f, valuation) ispisuje vrednost formule pri datoj valuaciji promenljivih. Valuacije se navode u obliku [p1->v1, ..., pn->vn]. Prijaviti grešku ukoliko navedena valuacija nije dovoljna za određivanje vrednosti formule (Npr. value(p/\q, [p->T])).

```
f1 := (p => (q /\ ~p)) => q;
f2 := {f1} /\ ~q;
f3 := substitute(f2, p, T);
print(f3);                                ((T => (q /\ (~T))) => q) /\ (~q)
value(f1, [p->T, q->F]);                  T
```

- (a) Gramatikom opisati dati jezik i konstruisati leksički analizator. [4]
(b) Definirati hijerarhiju klasa (struktura) za predstavljanje iskaznih formula u obliku stabla. [12]
(c) Implementirati funkciju print. [3]
(d) Implementirati funkciju substitute. [6]
(e) Implementirati funkciju value. [10]