

Prevodioci i interpretatori - Mart 2004.

teorijski deo

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) Konstruisati gramatiku koja opisuje deklaraciju prostih, pokazivačkih i nizovskih promenljivih u C-u.
(b) Za nisku

```
char **a,*b[3];
```

napisati najdešnje izvodjenje i nacrtati drvo izvodjenja (*parse tree*).

- (c) Konstruisati potisni automat koji metodom sintaksne analize navise prihvata dati jezik i simulirati njegov rad na datom primeru.

Prevodioci i interpretatori - Mart 2004.

teorijski deo

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) Konstruisati gramatiku koja opisuje deklaraciju prostih, pokazivačkih i nizovskih promenljivih u C-u.
(b) Za nisku

```
char **a,*b[3];
```

napisati najdešnje izvodjenje i nacrtati drvo izvodjenja (*parse tree*).

- (c) Konstruisati potisni automat koji metodom sintaksne analize navise prihvata dati jezik i simulirati njegov rad na datom primeru.

Prevodioci i interpretatori - Mart 2004.

teorijski deo

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) Konstruisati gramatiku koja opisuje deklaraciju prostih, pokazivačkih i nizovskih promenljivih u C-u.
(b) Za nisku

```
char **a,*b[3];
```

napisati najdešnje izvodjenje i nacrtati drvo izvodjenja (*parse tree*).

- (c) Konstruisati potisni automat koji metodom sintaksne analize navise prihvata dati jezik i simulirati njegov rad na datom primeru.

Prevodioci i interpretatori - Mart 2004.

teorijski deo

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) Konstruisati gramatiku koja opisuje deklaraciju prostih, pokazivačkih i nizovskih promenljivih u C-u.
(b) Za nisku

```
char **a,*b[3];
```

napisati najdešnje izvodjenje i nacrtati drvo izvodjenja (*parse tree*).

- (c) Konstruisati potisni automat koji metodom sintaksne analize navise prihvata dati jezik i simulirati njegov rad na datom primeru.

3. (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 rekurzivnog spusta prevodi ispravnu uslovnu dodelu u odgovarajući if-iskaz. Npr. gornji primer treba prevesti u

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

3. (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 rekurzivnog spusta prevodi ispravnu uslovnu dodelu u odgovarajući if-iskaz. Npr. gornji primer treba prevesti u

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

3. (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 rekurzivnog spusta prevodi ispravnu uslovnu dodelu u odgovarajući if-iskaz. Npr. gornji primer treba prevesti u

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

3. (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 rekurzivnog spusta prevodi ispravnu uslovnu dodelu u odgovarajući if-iskaz. Npr. gornji primer treba prevesti u

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