/

Elektrotehnički fakultet u Beogradu

Potpis dežurnog

Predmet: Programski prevodioci 1
Nastavnik: dr Dragan Bojić, red. prof.
Asistenti: doc. dr Maja Vukasović
mast.inž. Mihajlo Ogrizović
mast.inž. Luka Hrvačević
Ispitni rok: Drugi kolokvijum u školskoj 2025/2026.
Datum: 17.1.2025.

Kandidat: _____

Broj Indeksa: _____ *Smer:* _____ *Sala:* _____

Kolokvijum traje 90 minuta.
Nije dozvoljeno korišćenje literature.
Prvih sat vremena nije dozvoljeno napuštati kolokvijum.

Zadatak 1 _____*/7*

Zadatak 2 _____*/7*

Zadatak 3 _____*/6*

Ukupno: _____*/20*

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Na pitanja odgovarati **čitko i precizno**. Srećno!

1) (7 poena)

a) [2] Za gramatiku na slici konstruisati karakteristični LR(0) automat, potisnu tabelu i kontrolnu tabelu LR(0) parsera. Naznačiti eventualne parserske konflikte.
b) [2] Izračunati poništive neterminale, FIRST i FOLLOW skupove za datu gramatiku.
c) [2] Konstruisati kontrolnu tabelu SLR(1) parsera. Naznačiti eventualne parserske konflikte.

Napomena: za maksimalan broj poena jasno u rešenju razdvojiti tačke a, b i c i za svaki zahtev naznačiti šta je šta u rešenju (na primer koja tabela je prikazana).

d) [1] Predložiti gramatiku koja opisuje isti jezik kao data gramatika, a može se parsirati LR(0) parserom.

1. <S> → b<A>
2. <S> → <X>a
3. <X> → ε
4. <A> → <X>

Rešenje:

2) (7 poena)

Zadata je gramatika:

1. <S> → <S>a
2. <S> → b<A>
3. <A> → b<B>
4. <A> → ε
5. <A> → c
6. <B> → d

a) [4] Transformisati datu gramatiku u LL(1) gramatiku pa zatim za transformisanu gramatiku napisati parser na bazi rekurzivnog spusta.
b) [3] Ukratko objasniti koje probleme u implementaciji parsera na bazi rekurzivnog spusta prave leva rekurzija i leva faktorizacija.

Rešenje:

3) (6 poena)

Data je gramatika koja opisuje jednostavan opis konfiguracione sekcije. Konfiguracija se sastoji od imena sekcije i liste parametara. Svaki parametar ima ime i vrednost. Vrednost može biti numerička ili tekstualna. Ime sekcije i ime parametra predstavljeni su identifikatorom ID. Numerička vrednost je predstavljena terminalom NUM, dok je tekstualna vrednost predstavljena terminalom STR.

1. <conf> → <conf> <section>
2. <conf> → <section>
3. <section> → ID { <param_list> }
4. <param_list> → <param_list> <param>
5. <param_list> → <param>
6. <param> → ID = <value> ;
7. <value> → NUM
8. <value> → STR

a) [4] Dodeliti atribute i pravila računanja atributa datoj gramatici tako da se proveri semantička ispravnost konfiguracije prema sledećim pravilima: U okviru jedne sekcije ne smeju postojati dva parametra sa istim imenom. Parametar sa imenom port mora imati numeričku vrednost. Sekcija mora sadržati bar jedan parametar. U slučaju kršenja bilo kog od navedenih pravila, treba detektovati semantičku grešku tako što će se neterminalu conf proslediti sintetizovani atribut koji ima vrednost *false* u slučaju neispravnog izraza, odnosno *true* u suprotnom. Pretpostaviti da terminal ID ima sintetizovani atribut name, a terminali NUM i STR imaju sintetizovani atribut type čija vrednost može biti num ili str. Prilikom definisanja pravila za računanje simbola može se koristiti *if-else* struktura opisana u primeru.

   Primer ispravnog zapisa:
   server { host = localhost; port = 8080; } db { user=admin; password= 123; }
   Primer upotrebe nedozvoljenih zapisa:
   server { port = "abc"; }
   db { user = admin; user = root; }
   Primer if-else strukture: *atribut* ← *if (uslov) value1 else value2*

b) [2] Nacrtati atributivno stablo izvođenja za ulaz: server { port = 8080; }.

Rešenje:

Dodatni prostor za rad: