

Skripta sa zadacima za pripremu iz predmeta
Prevođenje programskih jezika

Pripremili:

Ivan Žužak, Ivan Budiselić, Zvonimir Pavlić,
Dejan Škvorc, Miroslav Popović, Goran Delač

Datum posljednje izmjene:

19. prosinca 2011.

Sadržaj

Predgovor	3
1 Uvod	4
2 Leksička analiza	6
3 Sintaksna analiza	12
4 Semantička analiza	29
5 Potpora izvođenju ciljnog programa	34
6 Generiranje međukoda	39
7 Generiranje ciljnog programa	42
8 Priprema izvođenja ciljnog programa	44
9 Optimiranje	45
Literatura	46

Predgovor

Ova skripta zadataka namijenjena je kao dopunski nastavni materijal za predmet Prevođenje programskih jezika na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. U svojoj PDF inačici, skripta je besplatno dostupna na Web-stranici predmeta. Skripta je nastala 2011. godine sakupljanjem i uređivanjem zadataka i teorijskih pitanja iz ispita, međuispita i auditornih vježbi predmeta *Prevođenje programskih jezika* od 2008. godine nadalje i predmeta *Automati, formalni jezici i jezični procesori 2* od 1999. do 2008. godine. Posebno zahvaljujemo autorima zadataka bez kojih izdavanje ovakve skripte ne bi bilo moguće: Ivan Budiselić, Goran Delač, Ivan Gavran, Andro Milanović, Miroslav Popović, Daniel Skrobo, Dejan Škvorc, Ivan Žužak.

Zadaci su u skripti organizirani u poglavlja, pri čemu naslovi poglavlja odgovaraju naslovima poglavlja u udžbeniku *Prevođenje programskih jezika* [1]. Uz zadatke je naznačen i dio udžbenika u kojem se obrađuje dio gradiva nužan za rješavanje zadatka i je li sličan zadatak riješen u sklopu auditornih vježbi. Rad na skripti nastavlja se i dalje, ali i u svom trenutnom obliku ona može biti korisna za savladavanje gradiva predmeta.

Skripta je oblikovana pomoću sustava L^AT_EX. S ciljem unaprjeđenja kvalitete skripte, izvorni L^AT_EX kod skripte javno je dostupan na adresi <http://github.com/fer-ppj/ppj-zbirka>. Molimo korisnike skripte da uočene pogreške ili primjedbe dojavu na službenu e-mail adresu predmeta ppj@zemris.fer.hr ili na stranicu za primjedbe: <http://github.com/fer-ppj/ppj-zbirka/issues>.

Zahvaljujemo Vedrani Janković na komentarima i prijavama pogrešaka pri pripremi ove inačice skripte. Na prijavama pogrešaka zahvaljujemo i Dini Šantlu, Marku Đuraseviću te Nini Uzelcu.

1. Uvod

1. Navedite prednosti i nedostatke uporabe jezičnih procesora. [1, str. 1]
2. Navedite tri jezika koji su vezani uz definiciju jezičnog procesora. [1, str. 3]
3. Navedite koja pravila se susreću kod programskih jezika te ih objasnite. [1, str. 4-11]
4. Objasnite samoprevodioca. [1, str. 27]
5. Objasnite cilj i osnovne korake postupka samopodizanja. [1, str. 27]
6. Objasnite što je funkcija preslikavanja jezičnog procesora i navedite njezine vrste. [1, str. 26]
7. Objasnite podjelu jezičnih procesora s obzirom na broj prolazaka kroz izvorni program. [1, str. 24-26]
8. Navedite i objasnite korake analize izvornog programa. [1, str. 3-13]
9. Objasnite razradbu jezičnih procesora s obzirom na dinamiku izvođenja procesa prevođenja. [1, str. 24-26]
10. Nabrojite i ukratko objasnite svaki od osnovnih koraka u izgradnji jezičnog procesora. [1, str. 3]
11. Koristeći Hoareov sustav oznaka CSP, opišite vrste jezičnih procesora s obzirom na dinamiku izvođenja. [1, str. 24]
12. Objasnite kompilatore i interpretatore, odnosno njihovu razliku, ne koristeći Hoareov sustav oznaka CSP. [1, str. 24-26]
13. Navedite faze rada jezičnog procesora od kojih se sastoje faza analize izvornog programa i faza sinteze ciljnog programa. [1, str. 3-4] [2]
14. Za svako od tri računala A , B i C s odgovarajućim strojnim jezicima a , b i c izgradite jezične procesore koji prevode jezik l u jezik n i koji su izvedivi na pojedinim računalima. Jezične procesore treba izgraditi pomoću raspoloživih jezičnih procesora $JP_l^{l \rightarrow n}$, $JP_a^{l \rightarrow a}$, $JP_b^{n \rightarrow o}$, $JP_o^{o \rightarrow b}$ i $JP_c^{o \rightarrow c}$. [1, str. 27]
15. Za računalno A postoji gotov jezični procesor $JP_a^{l \rightarrow a}$, dok je na računalu B na raspolaganju jezični procesor $JP_b^{n \rightarrow c}$. Računalno C novo je računalno i za njega ne postoji nikakav jezični procesor, a potrebno je program napisan u jeziku m prevesti u ciljni program za to računalno. Raspoloživ je i jezični procesor $JP_m^{m \rightarrow n}$, a zbog modularnosti je potrebno napisati jezični procesor $JP_l^{m \rightarrow l}$. Odredite u kojem je višem jeziku (l , m ili n) potrebno napisati taj jezični procesor da bi se na najkraći mogući način preveo program te navedite najkraći postupak prevođenja programa iz jezika m u ciljni jezik c . [1, str. 27] [2]
16. Tijekom razvoja složenog sustava koji se sastoji od tri računalne arhitekture A , B i C pojavila se potreba za razvojem jezičnog procesora $JP_b^{o \rightarrow b}$. Za potrebe sustava već su napisani jezični procesori $JP_a^{l \rightarrow a}$, $JP_c^{l \rightarrow n}$ i $JP_c^{o \rightarrow b}$. Osim toga, preko Interneta su javno dostupna još tri jezična procesora: $JP_l^{m \rightarrow b}$, $JP_m^{n \rightarrow o}$ i $JP_n^{o \rightarrow b}$. Budući da se projekt približava krajnjem roku, nema

dovoljno vremena za pisanje novog jezičnog procesora pa je potrebno jezični procesor $JP_b^{o \rightarrow b}$ konstruirati pomoću raspoloživih jezičnih procesora. Napišite postupak konstrukcije. [1, str. 27] [2]

17. Za računalu A postoji jezični procesor $JP_a^{z \rightarrow x}$, dok je na računalu B dostupan jezični procesor $JP_b^{x \rightarrow a}$. Raspoloživ je i jezični procesor $JP_z^{x \rightarrow y}$. Odredite u kojem višem programskom jeziku (x , y ili z) treba izgraditi jezični procesor $JP_{?}^{y \rightarrow b}$, tako da se može ostvariti prevođenje programa napisanog u jeziku x u ciljni jezik b . Navedite sve korake u postupku prevođenja programa. [1, str. 27] [2]
18. Računalni sustav sastoji se od jednog računala arhitekture A i jednog računala arhitekture B . Za računalnu arhitekturu B razvijen je jezični procesor $JP_b^{c \rightarrow b}$ kojim je omogućeno izvođenje programa napisanih višim programskim jezikom c na računalu arhitekture B . Osim toga, na raspolaganju su jezični procesori $JP_p^{c \rightarrow b}$, $JP_p^{p \rightarrow b}$, $JP_c^{p \rightarrow a}$ i $JP_c^{c \rightarrow b}$. Uporabom navedenih jezičnih procesora izgradite jezični procesor koji će na računalu arhitekture A omogućiti prevođenje programa napisanih višim programskim jezikom p u programe izvodive na računalu arhitekture B . [1, str. 27] [2]
19. Prikažite postupak izgradnje izvodivog jezičnog procesora koji prevodi jezik l u strojni jezik a . Na raspolaganju su samoprevodilac koji prevodi jezik l u strojni jezik a , $JP_a^{p \rightarrow s}$, $JP_a^{s \rightarrow a}$, $JP_b^{l \rightarrow q}$, $JP_b^{r \rightarrow a}$ i $JP_r^{q \rightarrow p}$ te računala A i B na kojima se mogu izvoditi programi pisani strojnim jezicima a i b . [1, str. 27] [2]
20. Tijekom razvoja složenog sustava koji se sastoji od dvije računalne arhitekture A i B pojavila se potreba za razvojem jezičnog procesora $JP_b^{l \rightarrow a}$. Za potrebe sustava već je napisan jezični procesor $JP_a^{m \rightarrow a}$. Osim toga, preko Interneta su javno dostupna još četiri jezična procesora: $JP_m^{l \rightarrow b}$, $JP_l^{m \rightarrow a}$, $JP_l^{l \rightarrow a}$ i $JP_m^{m \rightarrow b}$. Budući da se projekt približava krajnjem roku, nema dovoljno vremena za pisanje novog jezičnog procesora pa je potrebno jezični procesor $JP_b^{l \rightarrow a}$ konstruirati pomoću raspoloživih jezičnih procesora. Napišite postupak konstrukcije $JP_b^{l \rightarrow a}$. [1, str. 27] [2]
21. Za novo računalo “Amiga 2001” (računalo C) kao osnovni jezik treba upotrijebiti “C++” (jezik m) pa je potrebno izgraditi jezični procesor $JP_c^{m \rightarrow c}$. Na “PC” računalu (računalo B) postoji jezični procesor $JP_b^{l \rightarrow b}$, koji prevodi jezik “C” (jezik l) u Intelov strojni jezik te jezični procesor $JP_b^{k \rightarrow c}$, koji prevodi jezik “BCPL” (jezik k) u strojni jezik računala “Amiga 2001”. Na raspolaganju je i računalo “Amiga 4000” (računalo A) sa pripadnim jezičnim procesorom $JP_a^{k \rightarrow a}$, koji prevodi jezik “BCPL” u Motorolin strojni jezik. Pored ovoga postoje i jezični procesori $JP_k^{l \rightarrow k}$ (kros-kompilator) koji prevodi jezik “C” u jezik “BCPL” i koji je napisan u “BCPL”-u te $JP_l^{m \rightarrow c}$ koji prevodi “C++” u strojni jezik računala “Amiga 2001”, a napisan je u “C”-u. Napišite postupak samopodizanja kojim se može dobiti traženi jezični procesor uz uporabu postojećih jezičnih procesora. [1, str. 27] [2]

2. Leksička analiza

22. Opišite dinamiku izvođenja leksičke analize. [1, str. 52]
23. Opišite program simulator leksičkog analizatora zasnovan na tablici prijelaza ε -NKA. [1, str. 60-62]
24. Opišite način izrade globalne tablice znakova. [1, str. 50-52]
25. Nabrojite i objasnite osnovne klase leksičkih jedinki. [1, str. 46-49]
26. Opišite program simulator leksičkog analizatora zasnovan na tablici prijelaza DKA. [1, str. 58-60]
27. Objasnite kako leksički analizator utvrđuje leksičku pogrešku i navedite dva postupka oporavka od pogreške. [1, str. 58]
28. Navedite namjenu programa Lex i opišite strukturu ulazne datoteke Lexa. [1, str. 64-70]
29. Opišite podatkovne strukture leksičkog analizatora. [1, str. 49-52]
30. Objasnite metode opisa leksičkih jedinki, pravila određivanja klasa leksičkih jedinki i postupak grupiranja leksičkih jedinki koje se koriste za izgradnju generatora leksičkog analizatora. [1, str. 56-58]
31. Navedite i ukratko objasnite pet zadataka leksičkog analizatora. [1, str. 44-45]
32. Navedite i na primjeru objasnite dva osnovna načina razrješavanja nejednoznačnosti u leksičkoj analizi. [1, str. 63-64]
33. Navedite i objasnite varijable koje koristi simulator zasnovan na tablici prijelaza DKA. U ovisnosti o navedenim varijablama, objasnite postupak simulatora za grupiranje i određivanje klase leksičke jedinice. [1, str. 58-69]
34. Objasnite postupak razrješavanja nejednoznačnosti u leksičkoj analizi pretraživanjem lijevog konteksta. [1, str. 63-64]
35. Objasnite postupak razrješavanja nejednoznačnosti pretraživanjem desnog konteksta. [1, str. 63-64]
36. Navedite pravila za određivanje klase leksičke jedinice i grupiranje znakova u leksičke jedinice. [1, str. 57-58]
37. Navedite ulaze i izlaze iz generatora leksičkog analizatora i leksičkog analizatora ako je leksički analizator ostvaren kao zasebni prolaz jezičnog procesora. [1, str. 53]
38. Navedite strukture podataka pogodne za ostvarenje tablice znakova i asimptotsku složenost osnovnih operacija nad tablicom znakova za svaku predloženu strukturu podataka. [1, str. 50-52]
39. Opišite algoritam leksičkog analizatora zasnovanog na tablici prijelaza DKA. [1, str. 58-60]

40. Za zadani program nacrtajte sve tablice koje se stvaraju u leksičkoj analizi. Ključne riječi označene su masnim slovima. [1, str. 51] [2]

```
program Phoebe;
  j:=1; k:=1;
  ispiši (j,k);
  za i:=3 do 20 čini
    k:=j+k;
    j:=k-j;
    ispiši (k);
  kraj
kraj
```

41. Zadan je program u jeziku čije su ključne riječi, operatori i specijalni znakovi: **main**, **if**, **for**, **(**, **)**, **{**, **}**, **;**, **>**, **<**, **=**, **+**, **-**, *****. Ispišite sve tablice koje su izlaz leksičkog analizatora. [1, str. 51] [2]

```
main()
{
  a23=c+b57*27-3*a5;
  if(a>77)
  {
    a=a23;
    a=a+c;
  }
  for(i=0;i<12;i=i+1)
  {
    a=a+i;
    b57=a5-3.7*a5;
    i=i+1;
  }
}
```

42. Konstruirajte sve izlazne tablice leksičkog analizatora za dani izvorni program. Ključne riječi u programu masno su otisnute, a konstante su u kurzivu. [1, str. 51] [2]

```
import javax.swing.*;
public class FrameDemo {
  public static void main(String[] args) {
    JFrame jframe=new JFrame("Example");
    jframe.setSize(400,100);
    jframe.setVisible(true);
  }
}
```

43. Za zadani isječak kôda napišite sadržaje svih tablica leksičkog analizatora. Ključne su riječi masno otisnute. [1, str. 51] [2]

```
if (!fread(&id32, sizeof(struct ID3v2_Header),1,fp) ||
    strncmp(id32.id,"ID3",3))
return 0;
```

44. Napišite sve izlazne tablice simulatora leksičkog analizatora programskog jezika C za zadani isječak ulaznog programa. Pretpostavite da leksički analizator uspješno grupira višeznakovne operatore. [1, str. 51] [2]

Pozicija	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Znak	i	f	(i	f	3	=	=	2)	b	2	+	+	;

Pozicija	16	17	18	19	20	21	22	23	24	25
Znak	e	l	s	e	b	2	+	=	2	;

45. Leksički analizirajte zadani programski odsječak te konstruirati sve izlazne tablice leksičkog analizatora. Ključne su riječi masno otisnute. [1, str. 51] [2]

```
double factor = 0.356E-6;
int f1(const double &x) {
    if (!(x*factor< temp.treshold))
        return temp.high;
    return temp.low;
}
```

46. Konstruirajte sve izlazne tablice leksičkog analizatora za zadani programski odsječak napisan u programskom jeziku C. Ključne su riječi masno otisnute. [1, str. 51] [2]

```
clock_t clock (void)
{
    struct timeb now;
    clock_t elapsed;

    /* Calculate the difference between the initial time and now. */

    ftime(&now);
    elapsed=(now.time-_itimeb.time)*20;
    return(elapsed);
}
```

47. Leksički analizirajte zadani programski odsječak te konstruirati sve izlazne tablice leksičkog analizatora. Ključne su riječi podvučene, a sve su linije koje započinju znakom # komentari. [1, str. 51] [2]

```
def format_cond(string_list, c):
    # funkcija formatira ulaznu listu string_list ovisno o vrijednosti
    # parametra c
    format_func = lambda s: " ".join(s.split())
    for s in string_list:
        rez.append(format_func(s))
    return rez
```

48. Leksički analizirajte zadani programski odsječak te konstruirati sve izlazne tablice leksičkog analizatora. Ključne su riječi podvučene. [1, str. 51] [2]

```
const string Instrument = "Gitara";
enum Padez { Nominativ, Genetiv, Dativ, Akuzativ };
int a = (int) Padez.Nominativ + (int) Padez. Akuzativ;
Instrument += "Klasicna";
```


49. Za sljedeći niz izgradite tablicu uniformnih znakova, tablicu identifikatora i tablicu konstanti:
AKOI>31ONDAA++;INACEA+=4;

Ključne riječi i operatori, identifikatori te konstante definirani su izrazima:

```
KROS := < | > | AKO | ONDA | INACE | + | - | = | ;  
IDN := slovo ( slovo | brojka ) *  
KON := (brojka) +
```

Slovo i brojka redom označavaju velika i mala slova abecede te znamenke dekadskog sustava. Za razrješavanje nejednoznačnosti u grupiranju i određivanju klase leksičke jedinice koristite pravilo grupiranja najduljeg prefiksa koji je definiran barem jednim regularnim izrazom. Za prefikse jednake duljine, prednost ima izraz koji je zadan prvi. [1, str. 51] [2]

50. Ostvaren je program simulator leksičkog analizatora zasnovan na tablici prijelaza DKA s jednostavnim postupkom oporavka od pogreške. Simulator prepoznaje dva niza: **AUTO** i **AUTOMOBIL**. Na ulazu automata pojavljuje se niz **AUTOMATSKIAUTOMOBIL**. Odredite koje će nizove simulator leksičkog analizatora prepoznati i hoće li ispisati neke greške. Potrebno je i ispisati tablicu stanja unutarnjih kazaljki (početak, završetak i posljednji) programa simulatora za svaki učitani znak. [1, str. 58-60]
51. Generatoru leksičkog analizatora potrebno je zadati pravila za analizu Booleovih izraza. Booleovi se izrazi sastoje od identifikatora (počinju slovom), Booleovih konstanti (**TRUE** i **FALSE**), operatora (**&**, **|**, **^**, **~** i **=**) te okruglih zagrada. Napišite sve regularne izraze i akcije simulatora. [1, str. 47-48] [2]
52. U simulatoru leksičkog analizatora za aritmetičke izraze, potrebno je razriješiti nejednoznačnost uzrokovanu uporabom unarnih operatora. Aritmetički izrazi sastoje se od identifikatora (počinju slovom), cjelobrojnih konstanti, operatora **{+, -, *, / i =}** te okruglih zagrada. Napišite sve regularne izraze i akcije simulatora. [1, str. 47-48] [2]
53. Zadan je jezik L koji sadrži cjelobrojne aritmetičke izraze. Leksičke su jedinice jezika L varijable, konstante, binarni operatori **{+, -, *, /}**, okrugle zagrade i unarni operatori **{-, ++, --}**. Varijable se sastoje od slova i brojk i te moraju započinjati slovom, a konstante su cjelobrojne. Unarni operator **-** označava negaciju varijable ili konstante. Unarni operatori **++** i **--** imaju značenje kao u jeziku C. Definirajte pravila leksičkog analizatora kojima se ulazni niz rastavlja na leksičke jedinice. [1, str. 47-48]
54. U postupku sintaksne analize programskog jezika X potrebno je razlikovati dekadске, oktalne i heksadekadске pozitivne cjelobrojne konstante. Napišite regularne izraze koji će omogućiti ispravno određivanje klase cjelobrojne konstante u leksičkoj analizi. Oktalne konstante započinju znamenkom **0** (npr. **0134**, **071** i **00032**). Dopušteno je da dekadске konstante započinju vodećim nulama ako sadrže barem jednu znamenku **8** ili **9** (dekadske su konstante npr. **00039**, **488** i **455**). Heksadekadске konstante započinju nizom **0x** i dalje sadrže dekadске znamenke i mala slova **a, b, c, d, e, f** (npr. **0x13a**, **0x043f** i **0x00fed**). [1, str. 47-48]
55. Prikažite postupak obrade i izlaz leksičkog analizatora zasnovanog na regularnim izrazima iz tablice na sljedećim ulaznim nizovima (obrada svakog niza je nezavisna): [1, str. 47-48] [2]
- aabab
 - ababbba
 - abababc

r1	aab(c)*	ispiši("r1")
r2	(a)*b	ispiši("r2")
r3	abab	ispiši("r3")
r4	ab / c	ispiši("r4")
r5	ababb	uđi u stanje S; ODBACI;
r6	bbb	ispiši("r6")
r7	<S> bba	ispiši("r7"); izađi iz stanja S;
r8	(c)*	ispiši("r8")

56. Prikažite postupak obrade i izlaz leksičkog analizatora zasnovanog na regularnim izrazima iz tablice na sljedećim ulaznim nizovima (obrada svakog niza je nezavisna):

- 1) aababcaaab
- 2) ccababbba

Prikažite koji su znakovi ulaznih nizova uspješno grupirani pomoću kojih regularnih izraza. Nije potrebno prikazivati rad analizatora u smislu varijabli koje koristi analizator zasnovan na tablici prijelaza DKA ili ε -NKA. [1, str. 47-48] [2]

r1	aab(c)*	ispiši("r1")
r2	(a)*b	ispiši("r2")
r3	abab	ispiši("r3")
r4	ab / c	ispiši("r4")
r5	ababb	uđi u stanje S; ODBACI;
r6	bbb	ispiši("r6")
r7	<S> bba	ispiši("r7"); izađi iz stanja S;
r8	(c)*	ispiši("r8")

57. Prikažite postupak obrade i izlaz leksičkog analizatora zasnovanog na regularnim izrazima iz tablice na sljedećim ulaznim nizovima (obrada svakog niza je nezavisna):

- 1) aab
- 2) aaa%%b
- 3) b#%%

r1	%%b	ispiši("r1")
r2	%%ba*	ispiši("r2")
r3	a	ispiši("r3")
r4	aa / b	ispiši("r4")
r5	aaa	ispiši("r5")
r6	b	ispiši("r6")
r7	b(%)#	uđi u stanje S; ODBACI
r8	<S> #%%	ispiši("r8"); izađi iz stanja S

Prikažite koji su znakovi ulaznih nizova uspješno grupirani pomoću kojih regularnih izraza. Nije potrebno prikazivati rad analizatora u smislu varijabli koje koristi analizator zasnovan na tablici prijelaza DKA ili ε -NKA. [1, str. 47-48] [2]

58. Na osnovi navedenih pravila odredite i objasnite izlaz leksičkog analizatora za nizove a), b) i c). [1, str. 47-48] [2]

- a) yyy++x
- b) yyx
- c) x!++

r1	++x	ispiši("r1")
r2	++xy*	ispiši("r2")
r3	y	ispiši("r3")
r4	yy / x	ispiši("r4")
r5	yyy	ispiši("r5")
r6	x	ispiši("r6")
r7	x(+ !)	uđi u stanje S; ODBACI
r8	<S> !++	ispiši("r8"); izađi iz stanja S

59. Neka su zadana sljedeća pravila prevođenja u ulaznoj datoteci za program LEX:

```
%%
[a-z]+ ;
[0-4]+ ;
```

Za sljedeći ulazni niz prikažite izlaz leksičkog analizatora generiranog programom LEX za prethodno navedena leksička pravila: **Aaa5+Bbb4+ccc+DDD=91** [1, str. 64-70]

60. Za leksički analizator zasnovan na zadanom ε -NKA $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_0, \{q_3, q_4\})$ po koracima prikažite postupak računanja skupa ε -OKRUŽENJE($\delta(\{q_0, q_1\}, a)$) koristeći stog i dva bit-vektora. [1, str. 60-62]

δ	a	b	ε
q_0	$\{q_0, q_2\}$	$\{q_1, q_3\}$	$\{q_1, q_2\}$
q_1	$\{q_3\}$	-	$\{q_0, q_3\}$
q_2	-	-	$\{q_3\}$
q_3	$\{q_1, q_4\}$	-	-
q_4	-	$\{q_3, q_4\}$	-

3. Sintaksna analiza

61. Definirajte relacije *IspodZnaka* i *ReduciranZnakom* za parsiranje tehnikom *Pomakni-Pronađi*. [1, str. 121-123]
62. Navedite uvjete pod kojima je kontekstno neovisna gramatika ujedno i S-gramatika. [1, str. 87]
63. Navedite uvjete pod kojima je kontekstno neovisna gramatika ujedno i LL(1)-gramatika. [1, str. 98]
64. Navedite postupak izgradnje kanonskog LR-parsera na temelju izgrađenog DKA. [1, str. 150-151]
65. Objasnite postupak određivanja relacija prednosti na temelju zadane gramatike. [1, str. 133-135]
66. U pseudokodu sličnom jeziku C napišite algoritam parsiranja tehnikom prednosti operatora. [1, str. 130-133]
67. Navedite i kratko opišite postupke pretvorbe produkcija u produkcije LL(1)-gramatike. [1, str. 107-111]
68. Objasnite postupak lijevog izlučivanja za preuređivanje produkcija gramatike. [1, str. 108]
69. Objasnite postupak uklanjanja lijeve rekurzije tijekom pretvorbe produkcija u LL(1) oblik. [1, str. 110-111]
70. Navedite algoritam za izračunavanje ZAPOČINJE skupova za produkcije. [1, str. 102-103]
71. Opišite algoritam za izračunavanje relacije Ispred. [1, str. 105-107]
72. Navedite korake u računanju PRIMIJENI skupova za produkcije. [1, str. 107]
73. Opišite postupak računanja skupova SLIJEDI za prazne nezavršne znakove. [1, str. 107]
74. Navedite i kratko opišite podatkovnu strukturu sintaksnog analizatora. [1, str. 80]
75. Opišite kako se izvodi nadziranje i oporavak od pogrešaka kod LR-parsiranja. [1, str. 153-154]
76. Objasnite parsiranje od dna prema vrhu metodom *Pomakni-Reduciraj*. [1, str. 126-130]
77. Navedite i definirajte korake algoritma izgradnje kanonskog LR(1)-parsera. [1, str. 147-151]
78. Opišite postupak izgradnje potisnog automata za S-gramatiku. [1, str. 87-88]
79. Definirajte LL(1)-gramatiku i kratko opišite konstrukciju potisnog automata za LL(1)-gramatiku. [1, str. 95-99]
80. Navedite zadatke sintaksnog analizatora. [1, str. 71]
81. Navedite ulaze i izlaze iz sintaksnog analizatora ako je sintaksni analizator ostvaren kao zasebni prolaz jezičnog procesora. [1, str. 71]

82. Objasnite pojmove parsiranje, parsiranje od dna prema vrhu i parsiranje od vrha prema dnu. [1, str. 84,113]
83. Neovisno poredajte gramatike LL(1), S i Q te gramatike LALR(1), SLR(1), LR(0) i LR(1) uzlazno po općenitosti. [1, str. 71-154]
84. Objasnite namjenu programa Yacc te dijelova ulazne datoteke za program Yacc. [1, str. 154-158]
85. Ukratko objasnite postupak oporavka od pogreške kod LR-parsiranja koji se zasniva na traženju sinkronizacijskih znakova. [1, str. 153-154]
86. Navedite pet različitih vrsta sustava oznaka za opis sintaksnih pravila. [1, str. 81-95]
87. Opišite postupak sintaksne analize zasnovane na tablici Co-No. [1, str. 83-84]
88. Objasnite parsiranje od dna prema vrhu tehnikom *Pomakni-Reduciraj*. Opišite tablice koje se koriste u parsiranju. [1, str. 126-127]
89. Opišite algoritme na kojima se zasnivaju postupci oporavka od pogreške kod sintaksne analize. [1, str. 113,153]
90. Objasnite parsiranje od dna prema vrhu metodom prednosti operatora, relaciju prednosti, akcije parsera i određivanje uzorka za zamjenu. [1, str. 130-137]
91. Objasnite razlike u ostvarenju parsera LR(0), SLR(1), LALR i LR(1) te navedite njihove prednosti i nedostatke. [1, str. 138]
92. Objasnite konstrukciju ε -NKA u postupku izgradnje SLR(1)-parsera. [1, str. 147-149]
93. Navedite zahtjeve koje mora ispuniti detekcija pogrešaka u sintaksnom analizatoru. [1, str. 79]
94. Objasnite sustav oznaka COBOL. [1, str. 82]
95. Opišite postupak oporavka od pogrešaka u sintaksnom analizatoru. [1, str. 79-80]
96. Objasnite akcije parsera od dna prema vrhu koji koristi tehniku *Pomakni-Pronađi*. Opišite proturječja koja se pojavljuju. [1, str. 123-125]
97. Uklonite lijevu rekurziju iz sljedeće gramatike. [1, str. 110-111]

$$\begin{aligned} \langle S \rangle &\rightarrow a\langle A \rangle b\langle B \rangle a \mid b\langle B \rangle a\langle A \rangle b \\ \langle A \rangle &\rightarrow \langle A \rangle a\langle B \rangle b \mid \langle B \rangle a \mid a \\ \langle B \rangle &\rightarrow \langle A \rangle b \mid b \end{aligned}$$

98. Uklonite lijevu rekurziju iz dane gramatike. [1, str. 110-111]

$$\begin{aligned} \langle S \rangle &\rightarrow a\langle A \rangle b\langle B \rangle a \mid b\langle B \rangle a\langle A \rangle b \\ \langle A \rangle &\rightarrow \langle B \rangle b \mid b \\ \langle B \rangle &\rightarrow \langle A \rangle a \mid \langle B \rangle a\langle A \rangle b \mid a \end{aligned}$$

99. Uklonite lijevu rekurziju iz sljedeće gramatike. Odrediti je li dobivena gramatika LL(1). [1, str. 110-111]

$$\begin{aligned} \langle S \rangle &\rightarrow a\langle A \rangle b\langle B \rangle \\ \langle A \rangle &\rightarrow \langle B \rangle cd \mid a\langle A \rangle \mid \varepsilon \\ \langle B \rangle &\rightarrow \langle A \rangle b\langle S \rangle b \mid dc\langle S \rangle a \end{aligned}$$

100. Na sljedeću gramatiku primijenite algoritam razrješavanja lijeve rekurzije: [1, str. 110-111]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle a \langle A \rangle b \\ \langle A \rangle &\rightarrow \langle A \rangle a \mid \langle A \rangle b \mid c \langle A \rangle \mid d \langle B \rangle \mid f \\ \langle B \rangle &\rightarrow f\end{aligned}$$

101. Uklonite lijevu rekurziju iz zadane gramatike. [1, str. 110-111]

$$\begin{aligned}\langle S \rangle &\rightarrow a \langle A \rangle \langle B \rangle \mid b \langle B \rangle \\ \langle A \rangle &\rightarrow \langle A \rangle b \mid \langle B \rangle ac \mid a \\ \langle B \rangle &\rightarrow \langle A \rangle cb \mid bc\end{aligned}$$

102. Uklonite lijevu rekurziju iz zadane gramatike. Je li dobivena gramatika LL(1)? [1, str. 110-111]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle Z \rangle ef \mid ef \langle S \rangle \langle X \rangle \langle Z \rangle \\ \langle X \rangle &\rightarrow \langle Z \rangle ee \mid fg \langle S \rangle \\ \langle Z \rangle &\rightarrow \langle X \rangle ff \mid fe \langle X \rangle f \mid \varepsilon\end{aligned}$$

103. Za zadanu Q-gramatiku konstruirajte potisni automat. Tijekom parsiranja ulaznog niza na vrhu stoga redom se pojavljuju sljedeći znakovi: $\langle S \rangle$, $\langle A \rangle$, $\langle B \rangle$, $\langle B \rangle$, ∇ . Koji su se ulazni znakovi sigurno nalazili u parsiranom nizu? [1, str. 94-95] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow a \langle A \rangle \langle B \rangle \mid b \langle A \rangle \langle B \rangle \\ \langle A \rangle &\rightarrow a \langle A \rangle \mid b \langle A \rangle \mid c \mid \varepsilon \\ \langle B \rangle &\rightarrow d \mid e \langle B \rangle \mid f \langle A \rangle \mid \varepsilon\end{aligned}$$

104. Napišite gramatiku na temelju koje je konstruiran navedeni potisni automat te odredite o kojoj se gramatici radi. [1, str. 94-95] [2]

	a	b	\perp
S	1	2	8
A	3	4	8
B	5	6	8
a	7	8	8
b	8	7	8
∇	8	8	9

1. zamijeni aBbA; pomakni
2. zamijeni bAaB; pomakni
3. zamijeni bA; pomakni
4. zamijeni B; zadrži
5. izvuci; zadrži
6. zamijeni aB; pomakni
7. izvuci; pomakni
8. odbaci
9. prihvati

105. Za zadanu gramatiku konstruirajte konačni potisni automat i izrazite ga pomoću tablice. Prikažite rad potisnog automata na nizu **ebabc**. [1, str. 94-95] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle \langle B \rangle \mid d \langle C \rangle \\ \langle A \rangle &\rightarrow a \langle C \rangle \langle B \rangle \mid \varepsilon \\ \langle B \rangle &\rightarrow b \langle C \rangle \mid e \langle S \rangle \\ \langle C \rangle &\rightarrow a \langle B \rangle \mid c\end{aligned}$$

106. Iz navedenog potisnog automata rekonstruirajte gramatiku. [1, str. 94-95] [2]

	a	b	c	d	e	\perp
<S>	1	9	9	1	1	9
<A>	6	9	9	9	7	9
	2	9	9	3	2	9
<C>	4	5	9	9	9	9
∇	9	9	9	9	9	8

- 1: zadrži; zamijeni (d <A> b)
- 2: zadrži; zamijeni (d <A>)
- 3: pomakni; zamijeni (<C> c <S>)
- 4: pomakni; zamijeni (<C>)
- 5: zadrži; izvuci
- 6: pomakni; zamijeni (<A>)
- 7: pomakni; izvuci
- 8: prihvati
- 9: odbaci

107. Za zadanu gramatiku konstruirajte konačni potisni automat i izrazite ga pomoću tablice. Prikažite rad potisnog automata na nizu **cdabed**. [1, str. 94-95] [2]

<S> \rightarrow a<A>b | c<A>
 <A> \rightarrow a<A> | b | d<A>e<C>
 \rightarrow b | e<C>
 <C> \rightarrow c | d | e

108. Zadanu COBOL notaciju prevedite u BNF notaciju. [1, str. 81-82]

$$\left\{ \begin{array}{c} \underline{A} \\ \underline{B} \\ \underline{CD} \\ \underline{CE} \end{array} \right\} \left\{ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \right\} \dots \left\{ \begin{array}{c} \underline{A} \\ \underline{B} \\ \underline{DC} \\ \underline{EC} \end{array} \right\}$$

109. Zadanu gramatiku u BNF notaciji prevedite u COBOL notaciju. [1, str. 81-82]

<S> \rightarrow <D>CA | <D>CBA | <E>A
 <D> \rightarrow <D>D | ABC
 <E> \rightarrow <E>E | A

110. Sljedeće pravilo zapisano u COBOL notaciji napišite u BNF notaciji: [1, str. 81-82]

$$\underline{P} \left\{ \begin{array}{c} \left[\begin{array}{c} \underline{A} \\ \underline{N} \end{array} \right] \left\{ \begin{array}{c} \underline{N} \\ \underline{KB} \end{array} \right\} \\ \underline{TNK} \end{array} \right\} \dots$$

111. Navedeni izraz zapisan u COBOL notaciji pretvorite u BNF notaciju. [1, str. 81-82]

A[B]...[A[B]...]...C[C]...

112. Zadani BNF zapis pretvorite u COBOL zapis. [1, str. 81-82]

```

<if> → IF <izraz> THEN <blok>
<if> → IF <izraz> THEN <blok> ELSE <blok>
<blok> → N
<blok> → BEGIN <naredbe> END
<naredbe> → N
<naredbe> → N <naredbe>
<izraz> → A=B
<izraz> → A<>B
<izraz> → A<B
<izraz> → A<=B
<izraz> → A>B
<izraz> → A>=B

```

113. Zadanu BNF notaciju pretvorite u COBOL notaciju. [1, str. 81-82]

```

<S> → a<A> | b<B> | c<C>
<A> → ab<A> | bac<A> | c
<B> → a | bc<B> | c<C>ca
<C> → ab | ba

```

114. Zadani COBOL zapis pretvorite u BNF zapis. [1, str. 81-82]

$$\left\{ \begin{array}{c} \underline{C} \\ \underline{B} \\ \underline{L} \end{array} \right\} \dots [\underline{NOT}] \left\{ \begin{array}{c} 2 \\ \underline{TO} \dots \end{array} \right\} \underline{B} [\underline{NOT}] \dots$$

115. Zadani COBOL zapis pretvorite u istovjetni BNF zapis. [1, str. 81-82]

$$IF A \left\{ \begin{array}{c} = \\ \neq \\ < \\ > \end{array} \right\} B THEN \left\{ \begin{array}{c} N \\ BEGIN \ N \ [N] \ END \end{array} \right\} \left[ELSE \left\{ \begin{array}{c} N \\ BEGIN \ N \ [N] \ END \end{array} \right\} \right]$$

116. Sljedeću gramatiku zapisanu u BNF notaciji prevedite u COBOL notaciju: [1, str. 81-82]

```

<realni_broj> → <brojka> | <brojka> <rb1>
<rb1> → <brojka> | <brojka> <rb1> | . <rb2> | <rb3>
<rb2> → <brojka> | <brojka> <rb2> | <rb3>
<rb3> → E <rb4> | e <rb4>
<rb4> → + <rb5> | - <rb5> | <rb5>
<rb5> → <brojka> | <brojka> <rb5>
<brojka> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

117. Primjenom sustava oznaka BNF opišite konstante s pomičnom točkom. Na primjer, **-2.3e10**, **0.02** i **.2e-15** ispravno su zapisane konstante s pomičnom točkom. [1, str. 81]

118. Primjenom sustava oznaka COBOL opišite konstante s pomičnim zarezom. Npr. konstante oblika **-3.3e10**, **-02.0**. [1, str. 82]

119. BNF sustavom oznaka opišite BNF sustav oznaka. Za varijable koristite uniformni znak **VARIJABLA**, a za konstante uniformni znak **KONSTANTA**. [1, str. 81]

120. BNF sustavom oznaka opišite niz naredbi definicije varijabli tipa **int** u programskom jeziku C. Dopuštena imena varijabli su **a**, **b** i **c**. Početne su vrijednosti varijabli opcionalne, a mogu biti isključivo dekadski brojevi. Primjer jednog niza naredbi je: [1, str. 81]

```
int a, b=261;
int c;
int a=123;
```

121. Zadana je Co-No tablica. Odredite rezultat izvođenja zadanog programa. [1, str. 83-84] [2]
 $, 7 \rightarrow x, 12 \rightarrow y, x * y \rightarrow x, 3 \rightarrow z, x / z \rightarrow y, x - y / 2 \rightarrow x, x * 2 / y + z \rightarrow x, x * 6 - y + z * 2 - 9 * x \rightarrow y, x + y / 2 \rightarrow z,$

	,	\rightarrow	*	/	+	-
,	greška	dohvati	dohvati	dohvati	dohvati	dohvati
\rightarrow	spremi	greška	greška	greška	greška	greška
*	greška	pomnoži	pomnoži	pomnoži	pomnoži	pomnoži
/	greška	podijeli	podijeli	podijeli	podijeli	podijeli
+	greška	zbroji	zbroji	zbroji	zbroji	zbroji
-	greška	oduzmi	oduzmi	oduzmi	oduzmi	oduzmi

122. Primijenite zadanu Co-No tablicu u parsiranju sljedećeg niza: $, 3 + 5 * 2 \rightarrow R, 6 + 2 * 3 < R > 18$, Odredite vrijednost koja se na kraju rada parsera nalazi u akumulatoru. [1, str. 83-84] [2]

	,	\rightarrow	<	>	+	*
,		#1	#1	#1	#1	#1
\rightarrow	#2	#2	#2	#2	#2	#2
<		#3		#3	#3	#3
>		#4		#4	#4	#4
+		#5	#5	#5	#5	#5
*		#6	#6	#6	#6	#6

#1: dohvati operand u akumulator

#2: spremi vrijednost akumulatora u operand

#3: ako je operand manji od akumulatora, stavi ga u akumulator

#4: ako je operand veći od akumulatora, stavi ga u akumulator

#5: pribroji operand akumulatoru

#6: pomnoži akumulator s operandom

123. Zadanom Co-No tablicom parsirajte dva niza naredbi. Odredite prihvaća li se zadani niz zadanom Co-No tablicom, napišite generirani niz naredbi ciljnog programa i odredite vrijednosti varijabli **a**, **b** i **c** nakon izvođenja ciljnog programa.

a) ; $4 \rightarrow a$; $5 \rightarrow b$; $a + b * 10 \rightarrow c$;

b) ; $5 \rightarrow c$; $3 \rightarrow d$; $c * d$; $c / d \rightarrow a$;

Tablica sadrži akcije generatora ciljnog programa za stogovni stroj. Akcija **PUSH** stavlja na vrh stoga zadanu vrijednost ili vrijednost zadane varijable. Akcija **POP** skida podatak s vrha stoga i sprema ga u zadanu varijablu. Akcije **ADD**, **SUB**, **MUL** i **DIV** skidaju dva podatka s vrha stoga, izvode operaciju i stavljaju rezultat na vrh stoga. Akcija – označava grešku u ulaznom nizu. [1, str. 83-84] [2]

		Desni operator					
		;	+	-	*	/	\rightarrow
Lijevi operator	;	-	#1;	#1;	#1;	#1;	#1;
	+	-	#1; #2;	#1; #2;	#1; #2;	#1; #2;	#1; #2;
	-	-	#1; #3;	#1; #3;	#1; #3;	#1; #3;	#1; #3;
	*	-	#1; #4;	#1; #4;	#1; #4;	#1; #4;	#1; #4;
	/	-	#1; #5;	#1; #5;	#1; #5;	#1; #5;	#1; #5;
	\rightarrow	#6;	-	-	-	-	-

pri čemu su akcije u tablici:

- #1: PUSH
- #2: ADD
- #3: SUB
- #4: MUL
- #5: DIV
- #6: POP

124. Primijenite zadanu Co-No tablicu u parsiranju sljedećeg niza: $,x+y+z>w*x\rightarrow u\rightarrow v+z\rightarrow R,$

		Desni operator				
		,	\rightarrow	$>$	+	*
Lijevi operator	,		#1	#1	#1	#1
	\rightarrow	#2	#2			
	$>$		#3		#3	#3
	+		#4	#4	#4	#4
	*		#5	#5	#5	#5

pri čemu su akcije u tablici:

- #1: dohvati operand u akumulator
- #2: spremi vrijednost akumulatora
- #3: ako je operand veći od akumulatora, stavi ga u akumulator
- #4: pribroji operand akumulatoru
- #5: pomnoži akumulator s operandom [1, str. 83-84] [2]

125. Zadana je Co-No tablica:

		Desni operator					
		,	\rightarrow	+	-	*	/
Lijevi operator	,	greška	spremi	greška	greška	greška	greška
	\rightarrow	dohvati	greška	zbroji	oduzmi	moži	dijeli
	+	dohvati	greška	zbroji	oduzmi	moži	dijeli
	-	dohvati	greška	zbroji	oduzmi	moži	dijeli
	*	dohvati	greška	zbroji	oduzmi	moži	dijeli
	/	dohvati	greška	zbroji	oduzmi	moži	dijeli

Odredite vrijednosti varijabli **a**, **b** i **c** nakon izvođenja sljedećeg programa:

$,1\rightarrow a,2\rightarrow b,3\rightarrow c,a*b+c\rightarrow c,c-1/a\rightarrow b,c+b*a\rightarrow a,a*7-b+3*a-c*4+7\rightarrow c,$ [1, str. 83-84] [2]

126. Napišite Co-No tablicu za parsiranje aritmetičkih nizova koji sadrže operacije +, *, < i >. Kao separator koristi se , (zarez), a za pohranu vrijednosti u akumulator koristi se znak \rightarrow . Operator < uspoređuje sadržaj akumulatora s operandom i u akumulator stavlja manji od njih. Analogno vrijedi i za operator >, s tim da se u akumulator stavlja veći od njih. Na primjer, za $2<3$ u akumulatoru ostaje 2, za $4<3$ u akumulator se stavlja 3, za $2>3$ u akumulator se stavlja 3 te za $4>3$ u akumulatoru ostaje 4. Odredite vrijednost varijable **b** nakon parsiranja sljedećeg programa: $,1+2\rightarrow a,2*2<a>2\rightarrow b,$ [1, str. 83-84] [2]

127. Izgradite tablicu relacija prednosti za zadanu operatorsku gramatiku. [1, str. 133-135]

$\langle S \rangle \rightarrow \langle A \rangle a \langle S \rangle \mid \langle B \rangle$
 $\langle A \rangle \rightarrow \langle B \rangle c \langle D \rangle \mid \langle D \rangle d$
 $\langle B \rangle \rightarrow b$
 $\langle D \rangle \rightarrow c \langle S \rangle a$

128. Na temelju zadane operatorske gramatike izgradite tablicu relacija prednosti. Prikažite parsiranje niza $(a \vee a \wedge \neg a) \vee \neg a \wedge a$ pomoću izgrađene tablice. [1, str. 133-135]

$$\begin{aligned}
\langle E \rangle &\rightarrow \langle E \rangle \wedge \langle T \rangle \mid \langle T \rangle \\
\langle T \rangle &\rightarrow \langle T \rangle \vee \langle P \rangle \mid \langle P \rangle \\
\langle P \rangle &\rightarrow \langle N \rangle \mid \neg \langle N \rangle \\
\langle N \rangle &\rightarrow (\langle E \rangle) \\
\langle N \rangle &\rightarrow a
\end{aligned}$$

129. Binarni operator $\#$ lijevo je asocijativan i manje prednosti od binarnog operatora $\$$. Operator $\$$ desno je asocijativan. U izrazima je dopušteno korištenje okruglih zagrada, a varijable su označene završnim znakom a . Pripremite tablicu prednosti za parser zasnovan na prednosti operatora koji će parsirati opisane nizove. [1, str. 135-136]
130. Odredite PRIMIJENI skupove svih produkcija u zadanoj gramatici. Odredite je li gramatika LL(1) i obrazložite. [1, str. 100]

$$\begin{aligned}
\langle S \rangle &\rightarrow a\langle S \rangle b\langle S \rangle \mid c\langle D \rangle \langle A \rangle \mid \varepsilon \\
\langle A \rangle &\rightarrow \langle B \rangle c \mid ba\langle B \rangle \\
\langle B \rangle &\rightarrow a\langle D \rangle \mid de\langle C \rangle \mid e\langle C \rangle \\
\langle C \rangle &\rightarrow \langle D \rangle f \mid a\langle B \rangle c \\
\langle D \rangle &\rightarrow e\langle D \rangle \mid \varepsilon
\end{aligned}$$

131. Odredite PRIMIJENI skupove svih produkcija u zadanoj gramatici. Je li zadana gramatika tipa LL(1)? [1, str. 100]

$$\begin{aligned}
\langle S \rangle &\rightarrow a\langle S \rangle \mid b\langle X \rangle \\
\langle X \rangle &\rightarrow \langle Y \rangle \mid a \\
\langle Y \rangle &\rightarrow \langle S \rangle a\langle X \rangle \mid c \mid \varepsilon
\end{aligned}$$

132. Odredite PRIMIJENI skupove svih produkcija u zadanoj gramatici. Je li zadana gramatika tipa LL(1)? Obrazložite odgovor. [1, str. 100]

$$\begin{aligned}
\langle S \rangle &\rightarrow a\langle S \rangle c\langle S \rangle \langle Z \rangle \mid b\langle S \rangle c\langle Y \rangle c\langle S \rangle \\
\langle Y \rangle &\rightarrow a\langle Y \rangle \langle Z \rangle c \mid cbb\langle Z \rangle \\
\langle Z \rangle &\rightarrow \langle S \rangle a\langle Z \rangle \mid c\langle Z \rangle c\langle Y \rangle \mid \varepsilon
\end{aligned}$$

133. Odredite PRIMIJENI skupove svih produkcija u zadanoj gramatici. Je li zadana gramatika tipa LL(1)? Obrazložite odgovor. [1, str. 100]

$$\begin{aligned}
\langle S \rangle &\rightarrow \langle C \rangle c\langle E \rangle \mid b\langle A \rangle b\langle B \rangle \mid ded\langle A \rangle a \mid \varepsilon \\
\langle A \rangle &\rightarrow c\langle B \rangle f \mid f\langle B \rangle d \mid \varepsilon \\
\langle B \rangle &\rightarrow \langle C \rangle b \mid c\langle D \rangle \mid \varepsilon \\
\langle C \rangle &\rightarrow ad\langle C \rangle d \mid e\langle E \rangle c\langle D \rangle \langle A \rangle d \mid \varepsilon \\
\langle D \rangle &\rightarrow a\langle S \rangle fe \mid b\langle B \rangle de \mid \varepsilon \\
\langle E \rangle &\rightarrow da\langle C \rangle \mid e\langle A \rangle e\langle S \rangle fa
\end{aligned}$$

134. Konstruirajte potisni automat za zadanu LL(1)-gramatiku te odredite koji su mogući ulazni nizovi ako su se tijekom parsiranja na vrhu stoga redom pojavili nezavršni znakovi $\langle S \rangle \langle S \rangle \langle A \rangle \langle A \rangle \langle B \rangle \langle B \rangle$ (namjerno nije navedeno pojavljivanje završnih znakova na vrhu stoga). [1, str. 99]

$$\begin{aligned}
\langle S \rangle &\rightarrow a\langle S \rangle c \mid b\langle A \rangle d \\
\langle A \rangle &\rightarrow \langle B \rangle \mid b\langle A \rangle d \\
\langle B \rangle &\rightarrow e\langle B \rangle \mid \varepsilon
\end{aligned}$$

135. Odredite PRIMIJENI skupove za produkcije zadane gramatike (nije potrebno koristiti algoritam od 12 koraka). Odredite je li gramatika LL(1) i obrazložite odgovor. [1, str. 100]

$$\begin{aligned}\langle S \rangle &\rightarrow a\langle A \rangle \langle S \rangle b\langle C \rangle \langle D \rangle \mid \varepsilon \\ \langle A \rangle &\rightarrow c\langle A \rangle \langle S \rangle e \mid \varepsilon \\ \langle B \rangle &\rightarrow a\langle S \rangle fc \mid cba\langle B \rangle \mid \varepsilon \\ \langle C \rangle &\rightarrow ab\langle C \rangle d \mid c\langle C \rangle e \mid \varepsilon \\ \langle D \rangle &\rightarrow \langle A \rangle a\langle B \rangle f \mid d\langle B \rangle \langle D \rangle \mid \varepsilon\end{aligned}$$

136. Odredite PRIMIJENI skupove za produkcije dane kontekstno neovisne gramatike čiji je početni nezavršni znak $\langle A \rangle$. Odredite kojeg tipa je ta gramatika i navedite uvjete koje ispunjavaju gramatike tog tipa. [1, str. 85-115]

$$\begin{aligned}\langle A \rangle &\rightarrow a\langle E \rangle \mid c \mid \varepsilon \\ \langle B \rangle &\rightarrow b\langle C \rangle b\langle A \rangle \mid d \\ \langle C \rangle &\rightarrow a\langle B \rangle e \mid b \mid d\langle A \rangle b\langle D \rangle \\ \langle D \rangle &\rightarrow c\langle C \rangle f \mid \varepsilon \\ \langle E \rangle &\rightarrow a\langle D \rangle a \mid c\langle B \rangle f\end{aligned}$$

137. Zadanu gramatiku pretvorite u S-gramatiku. [1, str. 85] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow a\langle A \rangle b \mid b\langle A \rangle c \mid c\langle B \rangle a \\ \langle A \rangle &\rightarrow d \mid \varepsilon \\ \langle B \rangle &\rightarrow \langle B \rangle c \mid g\end{aligned}$$

138. Zadanu gramatiku pretvorite u S-gramatiku. [1, str. 85] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle C \rangle \mid a\langle A \rangle c \mid b\langle B \rangle a \\ \langle A \rangle &\rightarrow ab\langle A \rangle \mid bac\langle A \rangle \mid \varepsilon \\ \langle B \rangle &\rightarrow \langle C \rangle c \mid bc\langle B \rangle \mid \varepsilon \\ \langle C \rangle &\rightarrow cab \mid cba\end{aligned}$$

139. Zadanu gramatiku pretvorite u S-gramatiku koja generira isti jezik. [1, str. 85] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle C \rangle \mid a\langle A \rangle c \mid b\langle B \rangle a \\ \langle A \rangle &\rightarrow ab\langle A \rangle \mid bac\langle A \rangle \mid \varepsilon \\ \langle B \rangle &\rightarrow \langle C \rangle c \mid bc\langle B \rangle \mid \varepsilon \\ \langle C \rangle &\rightarrow cab \mid cba\end{aligned}$$

140. Pretvorite zadanu gramatiku u S-gramatiku. Provedite i postupak pojednostavljenja dobivene gramatike. [1, str. 85] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow ab\langle A \rangle \\ \langle A \rangle &\rightarrow \langle B \rangle bc \mid cb\langle A \rangle \\ \langle B \rangle &\rightarrow a\langle S \rangle a \mid cc\langle B \rangle b \mid \varepsilon\end{aligned}$$

141. Za zadanu kontekstno neovisnu gramatiku konstruirajte istovjetnu gramatiku koja nema ε -produkcija i u kojoj desna strana svake produkcije započinje završnim znakom. [1, str. 85] [2]

$$\begin{aligned}\langle A \rangle &\rightarrow a\langle B \rangle a \mid ac\langle B \rangle c \\ \langle B \rangle &\rightarrow \langle B \rangle a \mid a\langle A \rangle \mid \varepsilon\end{aligned}$$

142. Zadanu gramatiku pretvorite u Q-gramatiku i pokažite da dobivena gramatika zadovoljava pravila Q-gramatike. [1, str. 91-95]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle c \langle D \rangle \mid c \\ \langle A \rangle &\rightarrow \langle B \rangle a \mid b \langle D \rangle c \\ \langle B \rangle &\rightarrow \langle C \rangle e \mid ef\langle A \rangle \\ \langle C \rangle &\rightarrow \langle B \rangle \mid a \\ \langle D \rangle &\rightarrow a \langle D \rangle \mid b\end{aligned}$$

143. Zadanu Q-gramatiku pretvorite u S-gramatiku. [1, str. 85-95]

$$\begin{aligned}\langle S \rangle &\rightarrow a\langle A \rangle \langle C \rangle c \mid b\langle B \rangle \langle A \rangle a \\ \langle A \rangle &\rightarrow d\langle C \rangle \mid \varepsilon \\ \langle B \rangle &\rightarrow f \mid \varepsilon \\ \langle C \rangle &\rightarrow e\langle B \rangle\end{aligned}$$

144. Zadanu gramatiku pretvorite u LL(1)-gramatiku koja generira isti jezik: [1, str. 107-111]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle S \rangle ca \mid a\langle A \rangle c \mid ab\langle S \rangle b \\ \langle A \rangle &\rightarrow c\langle A \rangle c \mid d\langle B \rangle ac \mid db\langle A \rangle c \\ \langle B \rangle &\rightarrow baa\langle B \rangle \mid bbc\langle B \rangle \mid d\end{aligned}$$

145. Zadanu gramatiku prevedite u Q-gramatiku, uz pretpostavku da su **kon** i **var** završni znakovi. [1, str. 91-98]

$$\begin{aligned}\langle E \rangle &\rightarrow \langle E \rangle + \langle T \rangle \mid \langle T \rangle \\ \langle T \rangle &\rightarrow \langle P \rangle \mid \langle T \rangle * \langle P \rangle \\ \langle P \rangle &\rightarrow (\langle E \rangle) \mid \text{kon} \mid \text{var}\end{aligned}$$

146. Za zadanu gramatiku izgradite parser zasnovan na tehnici parsiranja *Pomakni-Pronađi*. [1, str. 121-123]

$$\begin{aligned}\langle S \rangle &\rightarrow b\langle A \rangle \mid p\langle A \rangle m\langle C \rangle \\ \langle A \rangle &\rightarrow d\langle S \rangle a \mid e \\ \langle C \rangle &\rightarrow d\langle A \rangle\end{aligned}$$

147. Odredite tip zadane gramatike i konstruirajte potisni automat parsera od vrha prema dnu. [1, str. 84-100]

$$\begin{aligned}\langle S \rangle &\rightarrow a\langle A \rangle b\langle B \rangle \mid b\langle B \rangle \langle A \rangle \\ \langle A \rangle &\rightarrow a\langle C \rangle \\ \langle B \rangle &\rightarrow b\langle A \rangle \mid \varepsilon \\ \langle C \rangle &\rightarrow cc\end{aligned}$$

148. Za zadanu gramatiku izgradite parser zasnovan na tehnici *Pomakni-Pronađi*. [1, str. 121-123]
[2]

$\langle S \rangle \rightarrow ab\langle A \rangle$
 $\langle A \rangle \rightarrow ab\langle B \rangle \mid bb$
 $\langle B \rangle \rightarrow c$

149. Za zadanu gramatiku konstruirajte parser zasnovan na tehnici *Pomakni-Pronađi*. [1, str. 121-123] [2]

$\langle S \rangle \rightarrow aba\langle S \rangle q\langle W \rangle \mid baf\langle Y \rangle$
 $\langle W \rangle \rightarrow cc\langle S \rangle \mid q$
 $\langle Y \rangle \rightarrow caba\langle W \rangle \mid ffa\langle S \rangle q\langle W \rangle$

150. Za zadanu gramatiku konstruirajte parser od dna prema vrhu metodom *Pomakni-Pronađi*. [1, str. 121-123] [2]

$\langle S \rangle \rightarrow a\langle S \rangle\langle A \rangle\langle B \rangle \mid b\langle B \rangle$
 $\langle A \rangle \rightarrow bad \mid c\langle A \rangle\langle B \rangle e$
 $\langle B \rangle \rightarrow ab \mid d\langle A \rangle e$

151. Za zadanu Q-gramatiku konstruirajte potisni automat. [1, str. 94-95] [2]

$\langle S \rangle \rightarrow ffg\langle A \rangle\langle B \rangle$
 $\langle A \rangle \rightarrow f\langle C \rangle ae\langle A \rangle\langle B \rangle \mid \varepsilon$
 $\langle B \rangle \rightarrow ga\langle A \rangle$
 $\langle C \rangle \rightarrow ee \mid g\langle S \rangle$

152. Konstruirajte potisni automat za zadanu gramatiku. [1, str. 94-95]

$\langle S \rangle \rightarrow a\langle B \rangle\langle A \rangle b \mid b\langle A \rangle\langle B \rangle c$
 $\langle A \rangle \rightarrow ba\langle B \rangle ba \mid bb\langle B \rangle bb \mid c\langle B \rangle c$
 $\langle B \rangle \rightarrow a\langle B \rangle a \mid \varepsilon$

153. Konstruirajte potisni automat koji prihvća nizove koje generira zadana gramatika. [1, str. 98-99]

$\langle S \rangle \rightarrow a\langle A \rangle b\langle B \rangle c \mid b\langle B \rangle a\langle A \rangle a$
 $\langle A \rangle \rightarrow \langle A \rangle c\langle B \rangle a \mid \langle B \rangle a \mid a$
 $\langle B \rangle \rightarrow \langle B \rangle b \mid b$

154. Konstruirajte potisni automat za zadanu Q-gramatiku. [1, str. 94-95] [2]

$\langle S \rangle \rightarrow a\langle A \rangle\langle B \rangle c \mid c\langle B \rangle\langle A \rangle b$
 $\langle A \rangle \rightarrow a\langle A \rangle \mid \varepsilon$
 $\langle B \rangle \rightarrow b\langle B \rangle \mid c$

155. Konstruirajte jednostavni potisni automat za navedenu LL(1)-gramatiku. Automat prikazite u tabličnom obliku. [1, str. 99]

$\langle S \rangle \rightarrow \langle B \rangle\langle B \rangle b \mid cba\langle A \rangle \mid \varepsilon$
 $\langle A \rangle \rightarrow ca\langle B \rangle \mid dc\langle A \rangle a$
 $\langle B \rangle \rightarrow a\langle A \rangle\langle S \rangle d \mid b\langle S \rangle d$

156. Prikažite stanje stoga i položaj glave za čitanje ulazne trake tijekom parsiranja niza abca primjenom zadanog potisnog automata. Koje je akcije potisnog automata potrebno pridružiti ćeliji s oznakom **X** da bi niz **abca** bio prihvatljiv? Obrazložite odgovor. [1, str. 90]

	a	b	c	\perp
S	#1	#2	-	-
A	-	-	#3	-
B	X	-	-	-
b	-	#5	-	-
∇	-	-	-	#6

#1: ZAMIJENI(Ab); POMAKNI;
 #2: ZAMIJENI(B); POMAKNI;
 #3: ZAMIJENI(B); POMAKNI;
 #5: IZVUCI; POMAKNI;
 #6: PRIHVATI;
 -: ODBACI;

157. Neka je zadana sljedeća S-gramatika:

$\langle S \rangle \rightarrow ab\langle R \rangle \mid b\langle R \rangle b\langle S \rangle$
 $\langle R \rangle \rightarrow a \mid b\langle R \rangle$

te neka je zadana sljedeća tablica prijelaza potisnog automata:

	a	b	\perp
$\langle S \rangle$	(1)	(2)	Prihvati
$\langle R \rangle$	(3)	(4)	-
b	-	(3)	-
∇	-	-	-

(1) Zamijeni($\langle R \rangle$ b a); Pomakni;
 (2) Zamijeni($\langle S \rangle$ b $\langle R \rangle$ b); Pomakni;
 (3) Izvuci; Pomakni;
 (4) Zamijeni($\langle R \rangle$);
 (-) Odbaci;

Utvrđite prihvaća li potisni automat s navedenom tablicom prijelaza jezik zadan navedenom S-gramatikom. Ako ne, navedite potrebne izmjene u tablici kako bi potisni automat prihvaćao jezik. [1, str. 85-89]

158. Odredite produkcije i vrstu gramatike na temelju koje je konstruiran sljedeći potisni automat. [1, str. 85-99] [2]

	a	b	c	\perp
S	1	2	3	8
A	2	4	8	4
B	5	4	6	8
b	8	7	8	8
∇	8	8	8	9

1: zamijeni (cA); pomakni
 2: zamijeni (S); pomakni
 3: zamijeni (bB); pomakni
 4: izvuci; zadrži
 5: zamijeni (AbB); pomakni

- 6: zamijeni (SS); pomakni
- 7: izvuci; pomakni
- 8: odbaci
- 9: prihvati

159. Odredite gramatiku na temelju koje je konstruiran sljedeći potisni automat. [1, str. 85-99] [2]

	1	2	3	4	\perp
<S>	1	2	7	1	2
<A>	3	7	7	4	7
	7	5	7	7	7
<C>	7	7	7	6	7
∇	7	7	7	7	8

- 1: zadrži; zamijeni (2B1A)
- 2: zadrži; izvuci
- 3: pomakni; zamijeni (A)
- 4: zadrži; zamijeni (C)
- 5: pomakni; izvuci
- 6: pomakni; zamijeni (2S)
- 7: odbaci
- 8: prihvati

160. Napišite gramatiku na temelju koje je nastao zadani potisni automat i odredite tip gramatike. [1, str. 85-99] [2]

	a	b	c	d	e	\perp
<S>	9	9	9	9	4	9
<A>	9	9	9	9	5	9
	9	9	9	9	6	9
<C>	1	1	9	9	7	7
<D>	9	9	9	9	3	9
<E>	7	7	2	2	7	7
e	9	9	9	9	8	9
∇	9	9	9	9	9	10

- 1: zamijeni (<C>); pomakni
- 2: zamijeni (<E><D>); pomakni
- 3: zamijeni (e<A>); pomakni
- 4: zamijeni (<A>); zadrži
- 5: zamijeni (<C>); zadrži
- 6: zamijeni (<E><D>); zadrži
- 7: izvuci; zadrži
- 8: izvuci; pomakni
- 9: odbaci
- 10: prihvati

161. Rekonstruirajte gramatiku iz koje je nastao sljedeći potisni automat [1, str. 85-99] [2]

	()	+	*	konst	\perp
<E>	1	2	2	2	3	2
<T>	2	2	4	5	2	5
<P>	2	6	2	4	2	6
)	2	7	2	2	2	2
∇	2	2	2	2	2	8

- 1: Zamijeni(<T>)<E>; Pomakni;
- 2: Odbaci;
- 3: Zamijeni(<T>); Pomakni;
- 4: Zamijeni(<E>); Pomakni;
- 5: Zamijeni(<P>); Zadrži;
- 6: Izvuci; Zadrži;
- 7: Izvuci; Pomakni;
- 8: Prihvati;

162. Odredite produkcije gramatike na temelju koje je konstruiran sljedeći potisni automat. [1, str. 85-99] [2]

	a	b	c	\perp
S	#1	#2	#3	#8
A	#2	#4	#8	#4
B	#5	#4	#6	#8
b	#8	#7	#8	#8
∇	#8	#8	#8	#9

- #1: Zamijeni (A); Pomakni;
- #2: Zamijeni (S); Pomakni;
- #3: Zamijeni (bB); Pomakni;
- #4: Izvuci; Zadrži;
- #5: Zamijeni (AbB); Pomakni;
- #6: Zamijeni (SS); Pomakni;
- #7: Izvuci; Pomakni;
- #8: Odbaci;
- #9: Prihvati;

163. Nacrtajte DKA na temelju kojeg je izgrađen zadani LR-parser. Pomoću zadanog LR-parsera parsirajte niz: **cbacbaba**. [1, str. 140-144]

	Akcija				Novo stanje		
	a	b	c	\perp	S	A	B
0			p3			s1	
1		p2					
2				r1			
3		p4					
4	p7		p6				s5
5		r2					
6		r3					
7					s8		
8		p9					
9		r4					

- pX = Pomakni (X);
r1 = Reduciraj ($S \rightarrow Ab$); Prihvati ();

$r2 = \text{Reduciraj } (A \rightarrow cbB);$
 $r3 = \text{Reduciraj } (B \rightarrow c);$
 $r4 = \text{Reduciraj } (S \rightarrow aAb);$
 $sX = \text{Stavi } (X);$

164. Prikažite korake tijekom parsiranja niza **bbbc** primjenom zadanog LR(1)-parsera. [1, str. 141-144] [2]

Stanje	Akcija			Novo stanje		
	b	c	\perp	<S>	<A>	
0	P1					
1	P2				S3	
2	P5					S6
3	P4					S7
4			R3			
5		R3				
6		P8				
7			Prihvati			
8	R2					

$R1 = \text{Reduciraj } (<S> \rightarrow b <A>)$
 $R2 = \text{Reduciraj } (<A> \rightarrow b c)$
 $R3 = \text{Reduciraj } (\rightarrow b)$

165. Izgradite SLR(1)-parser za zadanu gramatiku. [1, str. 139-147] [2]

$<S> \rightarrow <S><S>* \mid <S><S>+ \mid c \mid v$

166. Za zadanu gramatiku konstruirajte SLR(1)-parser. [1, str. 139-147] [2]

$<S> \rightarrow m \mid q<P>$
 $<P> \rightarrow <Q><P>$
 $<Q> \rightarrow b$

167. Konstruirajte SLR(1)-parser za zadanu gramatiku. [1, str. 139-147] [2]

$<S> \rightarrow <A>c<C>e$
 $<A> \rightarrow a \mid b$
 $ \rightarrow <A> \mid \varepsilon$
 $<C> \rightarrow df<C> \mid \varepsilon$

168. Za zadanu gramatiku konstruirajte SLR(1)-parser. [1, str. 139-147] [2]

$<S> \rightarrow <A> \mid a<S>$
 $<A> \rightarrow a<A>a \mid ac$
 $ \rightarrow b<A>$

169. Konstruirajte SLR(1)-parser za zadanu gramatiku. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle B \rangle a \mid a \langle A \rangle \langle S \rangle \\ \langle A \rangle &\rightarrow \langle A \rangle c \mid cb \\ \langle B \rangle &\rightarrow a \langle S \rangle b \mid b \langle A \rangle\end{aligned}$$

170. Za zadanu gramatiku izgradite SLR(1)-parser. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow a \langle A \rangle c \\ \langle A \rangle &\rightarrow x \langle S \rangle \mid \varepsilon\end{aligned}$$

171. Konstruirajte SLR(1)-parser za zadanu gramatiku. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle B \rangle \mid a \langle B \rangle b \langle A \rangle \\ \langle A \rangle &\rightarrow \langle B \rangle b \langle A \rangle \mid c \\ \langle B \rangle &\rightarrow d \mid e \langle S \rangle \langle C \rangle \\ \langle C \rangle &\rightarrow f \langle S \rangle \mid \varepsilon\end{aligned}$$

172. Izgradite SLR(1)-parser za zadanu gramatiku. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle a \langle B \rangle b \\ \langle A \rangle &\rightarrow a \\ \langle B \rangle &\rightarrow c \langle A \rangle \mid \varepsilon\end{aligned}$$

173. Za zadanu gramatiku izgradite SLR(1)-parser. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle B \rangle \langle A \rangle \\ \langle A \rangle &\rightarrow \langle B \rangle b \langle A \rangle \mid a \langle B \rangle \mid b \\ \langle B \rangle &\rightarrow c \langle A \rangle\end{aligned}$$

174. Izgradite LR(0)-parser za zadanu gramatiku. [1, str. 139-147] [2]

$$\langle S \rangle \rightarrow \langle S \rangle a \langle S \rangle \mid b \langle S \rangle c \mid d$$

175. Konstruirajte LR(1)-parser za zadanu gramatiku. [1, str. 147-152] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow a \langle A \rangle b \langle A \rangle \mid ba \langle A \rangle \\ \langle A \rangle &\rightarrow a \langle A \rangle \mid \varepsilon\end{aligned}$$

176. Izgradite LR(1)-parser za zadanu gramatiku. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow b \langle A \rangle \\ \langle A \rangle &\rightarrow \langle S \rangle a \mid \varepsilon\end{aligned}$$

177. Za zadanu gramatiku izgradite LR(1)-parser. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle S \rangle a \mid a \langle A \rangle b \\ \langle A \rangle &\rightarrow ab \langle A \rangle \mid b \langle B \rangle \langle B \rangle \\ \langle B \rangle &\rightarrow \langle A \rangle \langle B \rangle \mid ab\end{aligned}$$

178. Za zadanu gramatiku izgradite LR(1)-parser. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow b\langle A \rangle\langle B \rangle \\ \langle A \rangle &\rightarrow b\langle B \rangle c \\ \langle B \rangle &\rightarrow b\end{aligned}$$

179. Konstruirajte kanonski LR-parser (LR(1)) za zadanu gramatiku. [1, str. 139-147] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow a\langle A \rangle c \mid b\langle A \rangle d \\ \langle A \rangle &\rightarrow a\langle A \rangle \mid b\langle A \rangle \mid \varepsilon\end{aligned}$$

180. Izgradite LALR(1)-parser za zadanu gramatiku. [1, str. 155-156]

$$\begin{aligned}\langle S \rangle &\rightarrow \langle A \rangle a\langle B \rangle b \\ \langle A \rangle &\rightarrow a \\ \langle B \rangle &\rightarrow c\langle A \rangle \mid \varepsilon\end{aligned}$$

4. Semantička analiza

181. Proširite sljedeću gramatiku svojstvima i akcijskim znakovima tako da se dobije L-atributna prijevodna gramatika koja će računati dekadsku vrijednost rimskih brojeva. Početni znak gramatike $\langle S \rangle$ neka ima samo jedno i to izvedeno svojstvo. Konstruirajte potisni automat za dobivenu gramatiku. [1, str. 180-195] [2]

$$\begin{aligned}\langle S \rangle &\rightarrow V\langle A \rangle \mid I\langle B \rangle \\ \langle A \rangle &\rightarrow I\langle C \rangle \mid \varepsilon \\ \langle B \rangle &\rightarrow V \mid I\langle D \rangle \mid \varepsilon \\ \langle C \rangle &\rightarrow I\langle D \rangle \mid \varepsilon \\ \langle D \rangle &\rightarrow I \mid \varepsilon\end{aligned}$$

182. Zadana je gramatika koja generira aritmetičke izraze čiji su operandi razlomci zapisani u obliku **brojnik:nazivnik**. Pretvorite zadanu gramatiku u LL(1)-gramatiku. Dobivenu gramatiku proširite svojstvima i akcijskim znakovima tako da se dobije L-atributna gramatika koja izračunava i potom ispisuje vrijednost aritmetičkog izraza. Definirajte akcijske znakove pomoću: cjelobrojnog dijeljenja, množenja i zbrajanja, funkcije **zajnaz** koja izračunava zajednički nazivnik i funkcije **ispis** koja ispisuje razlomak.

$$\begin{aligned}\langle S \rangle &\rightarrow \langle E \rangle \\ \langle E \rangle &\rightarrow \langle E \rangle + \langle T \rangle \mid \langle T \rangle \\ \langle T \rangle &\rightarrow \langle P \rangle \mid \langle T \rangle * \langle P \rangle \\ \langle P \rangle &\rightarrow (\langle E \rangle) \mid \text{broj:broj}\end{aligned}$$

Znak **:** nije operator dijeljenja, već specijalni znak koji odvaja brojnik od nazivnika. U produkcijama je **broj** završni znak koji predstavlja cijeli broj. Niti u jednom trenutku razlomak se ne smije pretvoriti u realni broj, već se tijekom izračunavanja vrijednosti izraza razlomci promatraju kao par cjelobrojnih vrijednosti (brojnik i nazivnik). [1, str. 107-111, 180-183] [2]

183. Zadana je L-atributna gramatika koja izračunava vrijednost jednostavnih izraza koji se sastoje od operacija zbrajanja i množenja. Završni znak broj cjelobrojna je konstanta čije svojstvo odgovara vrijednosti dekadskog broja. [1, str. 180-198] [2]

a) odredite nasljedna i izvedena svojstva te odrediti redoslijed računanja svojstava i domene za računanje svojstava svake produkcije,

b) definirajte akcije potisnog automata i pokazati promjene na stogu potisnog automata za ulazni niz **6+2*4**,

c) napišite parser metodom rekurzivnog spusta; trenutni znak na ulazu je pohranjen u globalnu varijablu **znak**, a sljedeći znak se u tu varijablu čita pozivom globalne funkcije **noviznak()**; pretpostaviti da za izvođenje akcijskih znakova već postoje odgovarajući potprogrami: **Zbroji()** i **Pomnozi()**.

$S_a \rightarrow U_b T_{c,d}$	$c \leftarrow b, a \leftarrow d$
$T_{a,b} \rightarrow +U_c \{Zbroji\}_{d,e,f} T_{g,h}$	$d \leftarrow a, e \leftarrow c, g \leftarrow f, b \leftarrow h$
$T_{a,b} \rightarrow \varepsilon$	$b \leftarrow a$
$U_a \rightarrow P_b V_{c,d}$	$c \leftarrow b, a \leftarrow d$
$V_{a,b} \rightarrow *P_c \{Pomnozi\}_{d,e,f} V_{g,h}$	$d \leftarrow a, e \leftarrow c, g \leftarrow f, b \leftarrow h$
$V_{a,b} \rightarrow \varepsilon$	$b \leftarrow a$
$P_a \rightarrow (S_b)$	$a \leftarrow b$
$P_a \rightarrow broj_b$	$a \leftarrow b$

184. Zadanu gramatiku koja prihvaća oktalne brojeve s pomičnim zarezojima proširite svojstvima i akcijskim znakovima koji će izračunati dekadsku vrijednost pročitanoj broja (također s pomičnim zarezojima). Pretpostavite da završni znak **znamenka** ima jedno svojstvo koje je izvedeno i predstavlja vrijednost oktalne znamenke (0-7). [1, str. 177-180] [2]

$\langle S \rangle \rightarrow \text{znamenka } \langle A \rangle . \langle B \rangle$
 $\langle A \rangle \rightarrow \text{znamenka } \langle A \rangle \mid \varepsilon$
 $\langle B \rangle \rightarrow \text{znamenka } \langle B \rangle \mid \varepsilon$

185. Odredite koja su izvedena, a koja nasljedna svojstva te redoslijed računanja svojstava za sve produkcije zadane L-atributne gramatike. [1, str. 173-181]

$S_{a,b} \rightarrow x_c A_{d,e} \{Akc1\}_{f,g} B_{h,i,j} z_k \{Akc2\}_{m,n,o}$	$d \leftarrow a, e \leftarrow c, f \leftarrow c, h \leftarrow g,$ $m \leftarrow i, b \leftarrow i, n \leftarrow j, o \leftarrow k$
$A_{a,b} \rightarrow y_c z_d A_{e,f} \{Akc3\}_{g,h}$	$e \leftarrow a, f \leftarrow c, g \leftarrow b, h \leftarrow d$
$A_{a,b} \rightarrow A_{c,d} z_e \{Akc3\}_{f,g}$	$c \leftarrow a, d \leftarrow b, f \leftarrow d, g \leftarrow e$
$A_{a,b} \rightarrow x_c B_{d,e,f} x_g \{Akc4\}_{h,i,j,k,m}$	$d \leftarrow a, h \leftarrow b, i \leftarrow c, j \leftarrow e, k \leftarrow f, m \leftarrow g$
$B_{a,b,c} \rightarrow z_d x_e B_{f,g,h} \{Akc5\}_{i,j,k,m}$	$f \leftarrow e, i \leftarrow a, j \leftarrow d, k \leftarrow g, b \leftarrow h, c \leftarrow m$
$B_{a,b,c} \rightarrow y_d \{Akc6\}_{e,f,g} y_h \{Akc6\}_{i,j,k} y_m$	$e \leftarrow a, f \leftarrow d, i \leftarrow g, j \leftarrow h, b \leftarrow m, c \leftarrow k$
$B_{a,b,c} \rightarrow S_{d,e} x_f \{Akc6\}_{g,h,i}$	$d \leftarrow a, g \leftarrow e, h \leftarrow f, b \leftarrow e, c \leftarrow i$

186. Za zadanu L-atributnu gramatiku napišite parser zasnovan na rekurzivnom spustu. Koristite pseudokôd sličan programskom jeziku C. U kôdu koristite reference umjesto pokazivača te zanemariti deklaracije varijabli. Pretpostavite da su izlazne akcije već ostvarene kao zasebni potprogrami. Na raspolaganju su još i globalna varijabla **proc_znak** u kojoj je pohranjen zadnji pročitani ulazni znak, potprogram **sljij_znak()** koji učitava sljedeći ulazni znak te potprogram **greska()** koji zaustavlja parsiranje i ispisuje poruku o grešci. [1, str. 195-198]

$S_s \rightarrow \{Zbroji\}_{a,b,c} [L_{d,e}] \{Ispisi\}_{f,g}$	$s \leftarrow 1 \ a \leftarrow s \ b \leftarrow 1 \ e \leftarrow c \ f \leftarrow d \ g \leftarrow c$
$L_{a,b} \rightarrow \{Oduzmi\}_{c,d,e} [L_{f,g}] \{Ispisi\}_{h,i}$	$a \leftarrow 0 \ c \leftarrow b \ d \leftarrow 1 \ g \leftarrow e \ h \leftarrow f \ i \leftarrow e$
$L_{a,b} \rightarrow X_c L_{d,e} \{Oduzmi\}_{f,g,h}$	$a \leftarrow h \ e \leftarrow b \ f \leftarrow d \ g \leftarrow c$
$L_{a,b} \rightarrow \varepsilon$	$a \leftarrow 0$
$X_a \rightarrow a$	$a \leftarrow -1$

187. U pseudokodu sličnom jeziku C napišite parser zasnovan na metodi rekurzivnog spusta za zadanu gramatiku. [1, str. 195-198]

$\langle S \rangle \rightarrow a_p \langle B \rangle_{q,r} \langle A \rangle_{x,y} b_z \{Ispisi_w\}$	$q \leftarrow p + 3, x \leftarrow r, w \leftarrow z + y \times p$
$\langle A \rangle_{p,q} \rightarrow \varepsilon$	$q \leftarrow p \% 3$
$\langle B \rangle_{p,q} \rightarrow a_r \langle A \rangle_{x,y} b_z$	$x \leftarrow p, q \leftarrow y \times z + r$

188. Navedite sve ulazne nizove za koje zadana gramatika generira sljedeći niz izlaznih završnih znakova: {x}{q}{b}{q}{w}{a}. Za svaki ulazni niz nacrtajte sintaksno stablo. [1, str. 171]

$\langle S \rangle \rightarrow b \{z\} a \langle A \rangle \mid \langle A \rangle \{w\} b \{a\} a$
 $\langle A \rangle \rightarrow \{x\} \langle B \rangle c \langle B \rangle \langle A \rangle \{q\} \mid c \{y\} \{z\} a \langle A \rangle b \{p\} \mid a$
 $\langle B \rangle \rightarrow \langle B \rangle a \{q\} c b \{b\} \langle A \rangle \mid b a \{q\} \langle A \rangle c \{b\} \mid c$

189. Zadana je gramatika koja opisuje deklaraciju i inicijalizaciju dvodimenzionalnog polja. Odredite PRIMIJENI skupove za sve produkcije. Proširite gramatiku svojstvima i akcijskim znakovima tako da se omogući provjera ispravnosti deklaracije i inicijalizacije. Osim operatora `=, [,], {, }, ;, s_zarez`, u završne znakove spadaju i ključne riječi `kr_int`, `kr_char`, `kr_double`, identifikatori `idn` te konstante `const`. Pretpostavite da završni znak `const` ima jedno izvedeno svojstvo koje označava jedan od tri moguća tipa konstante. Veličina polja može se zadati samo pomoću cjelobrojnih konstanti. Tijekom provjere nije potrebno ispitivati ispravnost dimenzija inicijalizacijskog dijela. [1, str. 178-180]

```

<S> → <T> idn <D> = <I> ;
<D> → [ const ][ const ]
<E> → { <T> <K> }
<I> → { <E> <L> }
<K> → s_zarez <T> <K> | ε
<L> → s_zarez <E> <L> | ε
<T> → const | kr_char | kr_double | kr_int

```

190. Gramatiku koja služi za parsiranje naredbe deklaracije s pridruživanjem proširite svojstvima i akcijskim znakovima tako da se provjerava ispravnost tipa s lijeve i desne strane operatora pridruživanja. Nadalje, izračunajte i ispišite vrijednost konstante koja se zadanim izrazom pridružuje deklariranoj varijabli. Jezik ne dopušta implicitnu pretvorbu tipova. [1, str. 178-180] [2]

```

<S>→ <T> <I> = <E> ;
<T>→ int | float | double
<I>→ idn
<V>→ c_int | c_float | c_double
<E>→ <V> <L> | <C> ( <E> )
<L>→ <O> <E> <L> | ε
<C>→ <T> | ε
<O>→ + | -

```

Napomena: Produkcija $\langle E \rangle \rightarrow \langle C \rangle (\langle E \rangle)$ omogućava eksplicitnu promjenu tipa izraza $\langle E \rangle$.

191. Za zadanu atributnu gramatiku, u pseudokodu sličnom jeziku C napišite parser metodom rekurzivnog spusta. [1, str. 195-198]

$\langle S \rangle_o \rightarrow a \langle A \rangle_p bc \langle B \rangle_{q,r} \{Ispisi\}_w$	$p \leftarrow o \quad w \leftarrow q \quad r \leftarrow o$
$\langle S \rangle_o \rightarrow b \langle A \rangle_p \{Zbroji\}_{r,w,z}$	$p \leftarrow o \quad w \leftarrow o \quad r \leftarrow o$
$\langle A \rangle_o \rightarrow c \langle B \rangle_{p,q} \{Oduzmi\}_{r,w,z}$	$r \leftarrow o \quad w \leftarrow p \quad q \leftarrow o$
$\langle B \rangle_{o,p} \rightarrow ac$	$o \leftarrow p + 2$

192. Izgradite atributnu prijevodnu gramatiku koja parsira nizove cijelih dekadskih brojeva zapisane u obliku `{a1, a2, a3, ..., an}`.

Proširite gramatiku svojstvima i akcijskim znakovima koji računaju aritmetičku sredinu zadanog niza tako da sva pravila računanja budu pravila preslikavanja. Nizovi mogu biti proizvoljne duljine. [1, str. 178-180]

193. Izgradite atributnu prijevodnu gramatiku koja parsira nizove koji predstavljaju multiskup (skup u kojem se elementi mogu ponavljati) točaka u ravnini zapisan u obliku

`[(x1, y1), (x2, y2), ..., (xn, yn)]`.

U gramatici za broj koji predstavlja x ili y koordinatu točke koristiti završni znak **b**. Proširite gramatiku svojstvima i akcijskim znakovima koji računaju koordinate težišta multiskupa tako da sva pravila računanja budu pravila preslikavanja. [1, str. 178-180]

194. Opišite postupak izgradnje potisnog automata za prijevodnu gramatiku. [1, str. 184-195]
195. Objasnite što je provjera vrijednosti obilježja i opišite pojedine postupke za provjeru vrijednosti obilježja. [1, str. 200-202]
196. Opišite svojstva L-atributne prijevodne gramatike. [1, str. 180-181]
197. Nabrojite i objasnite formalne modele semantičkog analizatora. [1, str. 169-170]
198. Definirajte L-atributnu prijevodnu gramatiku. [1, str. 180-181]
199. Navedite i objasnite tri najčešće primjenjivana formalna modela semantičkog analizatora. [1, str. 169-170]
200. Objasnite sintaksom vođenu semantičku analizu. [1, str. 170-171]
201. Definirajte atributnu prijevodnu gramatiku. [1, str. 173]
202. Opišite algoritam provjere jednakosti tipova obilježja temeljen na provjeri jednakosti strukture obilježja. [1, str. 204-208]
203. Objasnite kako se obrađuju izvedena svojstva izlaznih završnih znakova koji se ne stavljaju na stog. [1, str. 173-176]
204. Navedite i objasnite algoritam ispitivanja jednakosti obilježja konstantnih vrijednosti. [1, str. 203-204]
205. Opišite korake gradnje atributnog generativnog stabla. Definirajte potpuno atributno generativno stablo. [1, str. 178]
206. Objasnite kako se obrađuju svojstva izvorišta koja nemaju dostupne vrijednosti. [1, str. 194]
207. Navedite uvjete pod kojima je atributna prijevodna gramatika ujedno i L-atributna prijevodna gramatika. [1, str. 180-181]
208. Objasnite razliku između izvedenih i nasljednih svojstava. Kako se izvedena i nasljedna svojstva spremaju na stog tijekom parsiranja od vrha prema dnu? [1, str. 173-177]
209. Navedite zadatke semantičkog analizatora. [1, str. 160]
210. Izgradite atributnu prijevodnu gramatiku koja računa zbroj elemenata polja pozitivnih cijelih brojeva. Polje je zapisano u sljedećem formatu:

$[x_1 \ \$ \ x_2 \ \$ \ x_3 \ \$ \ \dots \ \$ \ x_n]$

U gramatici za brojeve x koristiti završni znak **b**. Proširite gramatiku svojstvima i akcijskim znakovima koji računaju zbroj elemenata polja tako da sva pravila računanja budu pravila preslikavanja. Polje može biti proizvoljne veličine. [1, str. 177-180] [2]

211. Izgradite atributnu prijevodnu gramatiku koja parsira parove binarnih brojeva zapisane u obliku

$x_1 \ x_2 \ x_3 \ \dots \ x_n \ \diamond \ y_1 \ y_2 \ y_3 \ \dots \ y_m$
 $x_i, y_i \in \{0, 1\}$

Simbol \diamond predstavlja operator zbrajanja koji za neparne bitove oba broja uzima vrijednost **0**. Bitovi se broje od najmanje značajnoga prema najznačajnijem, počevši od nule. Na primjer:

$$\mathbf{01010010} \diamond \mathbf{1011011011} = \mathbf{01010000} + \mathbf{0001010001}$$

Proširite izgrađenu gramatiku svojstvima i akcijskim znakovima koji računaju rezultat primjene operatora \diamond izražen u dekadskom obliku. Brojevi mogu imati proizvoljan broj znamenaka. [1, str. 177-180] [2]

212. Izgradite potisni automat za zadanu atributnu prijevodnu gramatiku. Za sve akcije *Zamijeni* prikazati stanje na stogu neposredno prije i neposredno poslije primjene akcije. [1, str. 184-195] [2]

$$\begin{array}{ll} (1) \langle S \rangle \rightarrow a_p b_q \langle A \rangle_r \{X_v\} & v \leftarrow p \times q + r \\ (2) \langle A \rangle_p \rightarrow a_q \langle B \rangle_r & p \leftarrow q + r \\ (3) \langle B \rangle_p \rightarrow c_q & p \leftarrow q \end{array}$$

5. Potpora izvođenju ciljnog programa

213. Za svaki od četiri načina prenošenja parametara odredite stanja varijabli *j*, *k* i *l* te polja *i* nakon izvođenja zadanog programskog odsječka. [1, str. 243-252]

```
int i[3]={5,6,7},j=8,k=2,l=0;

divmod(int a,int b,int c,int d)
{
    c=a/b;
    d=a%b;
    l=a%b+1;
}

divmod(i[k],j,k,l);
```

214. Za dani programski odsječak odredite ispis ako se kod poziva potprograma koristi: [1, str. 243-252]
- a) razmjena vrijednosti
 - b) razmjena adresa (kod rekurzivnog poziva šalje se ista adresa koja je primljena kao parametar)
 - c) razmjena imena

```
f(a,b)
{
    ispiši(a,b);
    ako (b>=1)
    {
        b=b-1;
        a=a-1;
        f(b,a);
    }
}

glavni()
{
    cijeli x[3]={1,0,1};
    cijeli y=2;

    f(y,x[y]);
    ispiši(x[0],x[1],x[2],y);
}
```

215. Opišite algoritam gradnje lanca kazaljki nelokalnih imena i vektora dubine gniježđenja kod statičkog pravila djelokruga ugniježđenih procedura. [1, str. 239]

216. Objasnite povezivanje imena izvornog programa i objekata ciljnog programa te relaciju okoline i relaciju stanja. [1, str. 221-222]
217. Ukratko definirajte relaciju okoline i relaciju stanja. [1, str. 221-222]
218. Objasnite načine ostvarenja dinamičkog pravila djelokruga. [1, str. 241-242]
219. Opišite mehanizam povratne razmjene vrijednosti parametara procedura te navedite način ostvarenja. [1, str. 243-252]
220. Navedite i kratko objasnite postupke za određivanje djelokruga deklaracije nelokalnih imena. [1, str. 236-241]
221. Objasnite način ostvarenja statičkog pravila djelokruga nelokalnih imena ugniježđenih procedura. [1, str. 236-241]
222. Navedite osnovne načine razmjene ulazno/izlaznih parametara procedura i što se zapisuje u opisnik pozvane procedure prilikom pojedinog načina razmjene. [1, str. 243-244]
223. Objasnite pojmove djelokrug deklaracije i životni vijek pridruživanja imena. Što se događa sa životnim vijekom pridruživanja imena prilikom rekurzivnih poziva potprograma? [1, str. 233-234]
224. Objasnite djelokrug deklaracije i navedite moguća pravila definiranja djelokruga deklaracija. Pravila nije potrebno objašnjavati. [1, str. 233-234]
225. Objasnite vektor dubine gniježđenja i algoritam njegove izgradnje. [1, str. 239]
226. Za prikazani programski odsječak odredite ispis ako se kod poziva potprograma koristi: (a) razmjena vrijednosti, (b) razmjena adresa, (c) razmjena imena i (d) povratna razmjena vrijednosti. [1, str. 243-252] [2]

```

varijabla x=0, y=3, z=-1;
polje o[3]=10, o[4]=20;
  Racunaj(p, q, r) {
    z = p + x
    q = q + 1
    Ispisi(p, q, r);
    r = z + q
  }
{
  za x = 3 do 4 {
    Racunaj(o[x], o[3+x%2], z);
    Ispisi(x, y, z, o[3], o[4]);
  }
}

```

227. Za zadani program prikažite sadržaj opisnika procedura u trenutku neposredno prije izvođenja naredbe 05 ako se koristi: (a) statičko pravilo djelokruga, (b) dinamičko pravilo djelokruga. U oba slučaja objasnite tijek izvođenja programa i prikažite što će se ispisati kao posljedica naredbe 17. [1, str. 234-242] [2]

```

01  Glavni()
02    int y = 3;
03    def Z(a)

```

```

04     def X(x){
05         vrati x+y;
06     }
07     def Y(y){
08         ako y <= 4 onda
09             vrati Y(5);
10         inače
11             vrati X(y);
12     }
13 {
14     Y(a);
15 }
16 {
17     ispiši Z(y)
18 }

```

228. Za zadani program prikazite sadržaj opisnika procedura u trenutku prije izvođenja naredbe **07** ako se koristi: (a) statičko pravilo djelokruga, (b) dinamičko pravilo djelokruga. [1, str. 234-242] [2]

```

01 Glavni()
02     int y = 3;
03     def Z(a)
04         int r = 5
05         def X(x)
06             {
07                 vrati x+y+1;
08             }
09         def Y(y)
10             {
11                 vrati X(y)+1;
12             }
13     {
14         Y(a);
15     }
16 {
17     ispiši Z(y+1)
18 }

```

229. Prikazite i objasnite izvođenje sljedećeg programa ako se za poziv procedure koristi (a) razmjena vrijednosti, (b) razmjena adresa i (c) razmjena imena [1, str. 243-252] [2]

```

01 varijabla a = 0;
02 polje V = {7, 8}; // V[0]=7, V[1]=8
03 procedura Proc(x, y) {
04     x = y;
05     a = 1;
06     y = a;
07 }
08 { // ovo je glavni program
09     Proc(V[0], V[a]);
10     Ispiši(a, V[0], V[1]);
11 }

```

230. Za zadani program izgradite stablo aktiviranja procedura. [1, str. 228-229] [2]

```
01  Glavni()
02    X(a)
03    {
04        vrati a + 1;
05    }
06    Y(b, c)
07    {
08        vrati c - b/4;
09    }
10
11    Z(d, e)
12    {
13        dok (d <= e)
14        {
15            d = X(d);
16            e = Y(d, e);
17            Z(d, e);
18            if (d == 5)
19            {
20                d = 8;
21                e = 8;
22                dalje;
23            }
24        }
25    }
26  {
27      Z(3, 7)
28  }
```

231. Prikažite razliku između razmjene parametara primjenom mehanizma razmjene adresa i mehanizma razmjene imena na sljedećem programu: [1, str. 243-252] [2]

```
01  var x = 0
02  polje A = {10, 20} // A[0]=10, A[1]=20
03  P(a) {
04      x = 1
05      a = 100
06      Ispisi(A[0], A[1])
07  }
08  {
09      P(A[x])
10  }
```

232. Za zadani program prikažite vrijednosti globalnih i lokalnih varijabli tijekom izvođenja programa. Razmjena parametara procedura ostvaruje se primjenom mehanizma razmjene imena. [1, str. 243-252] [2]

```
01  varijabla x=0, y=3, z=-1;
02  polje o = {0, 0, 0, 10, 20};
```

```

03  Racunaj(p, q, r) {
04      z = p + x;
05      z = (q + 1) % 2 + 3;
06      Ispisi(p, x, r);
07      r = z + q;
08  }
09  {
10      za x = 3 do 4 {
11          Racunaj(o[x], o[3+x%2], z);
12          Ispisi(x, y, z, o[3], o[4]);
13      }
14  }

```

6. Generiranje međukoda

233. Navedite i kratko opišite linearne oblike međukôda. [1, str. 257-259]
234. Navedite osnovne razine međukoda i objasnite namjenu svake razine. [1, str. 254-255]
235. Objasnite graf zavisnosti. [1, str. 260-261]
236. Za zadani programski odsječak nacrtajte grafove zavisnosti. Grafove zavisnosti prikazite u ovisnosti o parametru X za sljedeće slučajeve: [1, str. 257-259] [2]
- a) $X=a$ (jezični pretprocesor zamjenjuje simboličko ime X varijablom a)
 - b) $X=b$ (jezični pretprocesor zamjenjuje simboličko ime X varijablom b)

```
a = b + 5;
dok ( X == 10 )
    b = X + 3;
X = 20;
```

237. Navedite oblike međukôda te za svaki oblik navedite primjere. [1, str. 255-261]
238. Za zadani program izgradite graf tijeka izvođenja. [1, str. 257-259] [2]

```
01      Učitaj(x)
02      p := 1.0
03  L1:  t := abs(p*p-x)
04      if t <= 1e-9 goto L2
05      p := avg(p, x/p)
06      goto L1
07  L2:  Ispiši(p)
```

239. Za zadani programski odsječak izgradite graf zavisnosti programa koji pokazuje četiri vrste zavisnosti podataka i zavisnosti upravljačkog tijeka izvođenja programa. [1, str. 257-259] [2]

```
a=3+b; c=d+a;
ako (b<c) {
    b=b+a; d=3/(b+2);
}
inače ako (b=c){
    b=b+a; d=3/(b+2);
}
c = a + d;
```

240. Za prikazani isječak programa nacrtajte graf zavisnosti upravljačkog tijeka i graf zavisnosti podataka. [1, str. 257-259] [2]

```

01  p = 1
02  i = 20;
03  a = i / 4
04  ako(i >= 3)
05  {
06      q = q + i/a;
07      ako(a<10)
08      {
09          a = q*p;
10      }
11      i = i-1;
12  }
13  p = 3 * a;

```

241. Opišite graf zavisnosti programa i navedite sve zavisnosti koje se njime opisuju. [1, str. 257-259]
242. Za zadani isječak programa nacrtajte graf zavisnosti upravljačkog tijeka i graf zavisnosti podataka. [1, str. 257-259] [2]

```

a=b+c;
c=a;
ako (b>c)
{
    c=b;
    a=b+a;
}
inače
{
    n=m+a;
    m=n+a;
}
n=b+a;

```

243. Za prikazani isječak programa nacrtajte graf zavisnosti upravljačkog tijeka i graf zavisnosti podataka. [1, str. 257-259] [2]

```

p = 1
i = 20;
a = i / 4
dok(i >= 3)
{
    q = q + i/a;
    if(a<10)
    {
        a = q*p;
    }
    i = i-1;
}
p = 3 * a;

```

244. Za dani program nacrtajte graf zavisnosti kako za upravljački tijek, tako i za sve vrste zavisnosti podataka.[1, str. 257-259] [2]


```

b:=b*3;
a:=b+d;
ako a<143 tada
    c:=a%8;
    a:=a/8;
d:=b+d;

```

245. Navedite tri oblika međukoda. [1, str. 255]

246. Za zadani program izgradite graf tijeka izvođenja. [1, str. 257-259] [2]

```

01      Input(n)
02      Input(p)
03      a0 := 2
04      if n <= 5 goto L1
05      if p > 5 goto L2
06 L1:  a1 := a0 + 3
07      a2 := a1 + p
08      a3 := a1 * n
09      goto Z
10 L2:  a2 := 3 * 3
11      Output (a3)
12      p := p + 1
13      goto L1
14 Z:   nop

```

247. Izgradite atributnu prijevodnu gramatiku koja generira troadresne naredbe za računanje aritmetičkih izraza koji sadrže operatore + i *. U izrazima je dopušteno korištenje zagrada. [1, str. 178-180] [2]

248. Na temelju zadane gramatike izgradite atributnu prijevodnu gramatiku za generiranje troadresnih naredbi koje ostvaruju programe napisane u jeziku zadane gramatike. [1, str. 178-180]

```

<S> → <N> <S> | ε
<N> → <Assign> | <If> | <While>
<If> → if <Exp> then <N> else <N> end
<While> → while <Exp> do <N> end
<Assign> → idn '=' <Exp> ';'
<Exp> → <Exp> ∧ <Exp>
<Exp> → <Exp> ∨ <Exp>
<Exp> → idn

```

249. Izgradite atributnu prijevodnu gramatiku koja generira troadresne naredbe za računanje logičkih izraza koji sadrže operator ∧, ∨ i ¬. [1, str. 178-180]

7. Generiranje ciljnog programa

250. Napišite tablice generatora ciljnog programa koji kao ulaz koristi sintaksno stablo čije čvorove čine četiri matematičke operacije (zbrajanje, oduzimanje, množenje i dijeljenje). Ciljno računalo posjeduje samo jedan registar. Nacrtajte sintaksno stablo za izraz $(a/b + (c+d) * (e-f)) * (g-h/i)$ i navedite redoslijed kojim generator obilazi čvorove. [1, str. 281-284]
251. Objasnite generiranje ciljnog programa na temelju postfiksno sustava oznaka. [1, str. 279-280]
252. Opišite postupak izrade adresa naredbama. [1, str. 268]
253. Opišite Chaitinov heuristički postupak za bojanje grafa zavisnosti simboličkih i stvarnih registara. [1, str. 273]
254. Opišite algoritam generiranja ciljnog programa na osnovi troadresnih naredbi. [1, str. 276-279]
255. Navedite elemente strukture generatora ciljnog programa. [1, str. 265]
256. Objasnite kako se dobiva i boji graf zavisnosti simboličkih i stvarnih registara te kako se dodjeljuju stvarni registri. [1, str. 269-273]
257. Za zadani isječak programa odredite izlaz generatora ciljnog programa za Motorolinu seriju procesora 68000. Na raspolaganju su 3 registra (**D0**, **D1** i **D2**), varijablu **k** nije moguće pohraniti u registre **D0** i **D1**, a memorija se ne smije koristiti. Pretpostavite da se varijabla **n** koristi u nastavku programa. Ne zahtijeva se uporaba algoritma bojanja grafova.

```
i=0;
j=3;
k=i+j;
m=k*2;
i=m-5;
j=4*j;
n=j-i;
k=m+i;
```

258. Za dani isječak izvornog programa generirajte postfiksni sustav oznaka i pritom definirati korištene operatore grananja. Na temelju dobivenog postfiksno sustava oznaka, prikazite sadržaj stoga tijekom generiranja ciljnog programa za neki od Motorolinih mikroprocesora. [1, str. 256-257, 279-280]

```
dok (i>j)
{
    ako (i>k)
        k=i+j-m;
    inače
        i=j-4;
}
```

259. Za zadanu gramatiku napišite tablice generatora ciljnog programa koji kao ulaz koristi sintaksno stablo. Ciljni program je strojni jezik ili za Motoroline ili za Intelove procesore. Operator $+$ označava zbrajanje, a unarni operator $!$ označava logičko invertiranje vrijednosti prema sljedećem pravilu zapisanom u C notaciji: $!x = (x != 0) ? 0 : 1$. [1, str. 281-283]

```

<Naredba> → <Varijabla> = <Izraz>
<Izraz> → <Izraz> + <Izraz>
<Izraz> → ! ( <Izraz> )
<Izraz> → <Varijabla>
<Varijabla> → a | b | c | d | e

```

260. Prikažite tablicu upravljanja za operaciju množenja za generiranje ciljnog programa na osnovi sažetog sintaksnog stabla ako generator raspolaže samo jednim registrom **R**. [1, str. 281-283]
261. Na temelju naredbe izvornog programa $(a+b*c)/(f*g-(d+e)/(h+k))$ nacrtajte sažeto sintaksno stablo i na temelju njega tablično prikažite generiranje ciljnog programa. [1, str. 261-262, 281-284]
262. Na temelju naredbe izvornog programa $a+b*c$ nacrtajte sažeto sintaksno stablo i na temelju njega tablično prikažite generiranje ciljnog programa. [1, str. 261-262, 281-284]
263. Na temelju naredbe izvornog programa $(a+b*c)/(f-g-h)$ nacrtajte sažeto sintaksno stablo i na temelju njega tablično prikažite generiranje ciljnog programa. [1, str. 261-262, 281-284]
264. Objasnite algoritam generiranja ciljnog programa na temelju troadresnih naredbi. Prikažite postupak generiranja troadresnih naredbi za sljedeće naredbe izvornog programa: [1, str. 262-263, 276-279] [2]

```

p = (a + b/c) + (d/e)-(d+e) × e
r = (p × c) + (d × e × 4)-(d+e) × e

```

265. Za zadani programski odsječak primijenite algoritam bojanja grafova kako biste ostvarili pridruživanje registara **D0-D5**. [1, str. 269-273]

```

ako (j > 3) {
    j := 23;
    i := 11;
    m := j + 5;
} inače {
    i := 17;
    j := m + 5;
    ako (k < 4) {
        k := i + 8;
        m := j { 3;
    } inače ako (k == 5) {
        k := i;
        i := 19;
        m := i + k * i;
    } inače {
        m := 1;
    }
}
i := 4 + 3+m;

```

8. Priprema izvođenja ciljnog programa

266. Navedite razradbu jezičnih procesora s obzirom na stupanj pripremljenosti ciljnog programa za izvođenje. [1, str. 286]

9. Optimiranje

- 267. Opišite što se nastoji utvrditi analizom toka podataka. [1, str. 301-302]
- 268. Objasnite razliku između strojno nezavisnog i strojno zavisnog optimiranja. [1, str. 294]
- 269. Nabrojite komponente koje čine analizu izvođenja programa. [1, str. 297]
- 270. Opišite postupak optimiranja petlji kod međukoda niže razine i ciljnog programa. [1, str. 311-312]
- 271. Opišite analizu tijeka izvođenja programa. [1, str. 298-301]
- 272. Nabrojite i kratko opišite postupke optimiranja međukoda srednje razine. [1, str. 316-317]

Bibliografija

- [1] Siniša Srbljić: *Prevođenje programskih jezika*, Element, 2007.
- [2] Daniel Skrobo, Ivan Žužak, Miroslav Popović: *Prevođenje programskih jezika - auditorne vježbe*, ZEMRIS, 2007.