

1. Kod SAT-riješavača GRASP, neka je konfliktno pridruživanje nastalo nakon *alternativnog* izbora vrijednosti varijabli pri grananju na razini 5 bilo:  $KP = \{x1 = 0@1, x5 = 0@2, x3 = 0@3, x7 = 1@3\}$ . U tom slučaju, događa se povratak na razinu:
- 3
  - 1
  - 4
  - 2
2. Zadan je ekspanzirani asinkroni produkt ( $\sigma^w$ ) s vrijednostima varijabli:  
 $x = 2; x = 1; x = 5; x = 4; x = 7; x = 9; x = 10; x = 13;$   
 $x = 12; x = 11;$   
 Odredite za koji od ponuđenih odgovora je LTL formula  $\neg p \wedge (Fp \rightarrow \text{"eventaully"} p)$  istinita
- $p \equiv (x > 64)$
  - $p \equiv (x < 0)$
  - $p \equiv (x \geq 256)$
  - $p \equiv (x == 13)$
3. Algoritam Chaff provodi pretragu iznova:
- Dodajući mali šum pri prvih nekoliko odluka o grananju.
  - Periodičkim množenjem svih brojeva literala s nekom konstantom manjom od 1
  - Algoritam Chaff ne provodi pretragu iznova
  - Uklanjanjem svih naučenih klauzula i postavljanjem promatranih literala na druge slučajne vrijednosti.
4. Uklonjen koda u Verilogu unutar bloka *always*, koja izgleda:  $a \leftarrow b$ ; provodi se:
- Zadržavanje varijable  $b$  u odnosu na varijablu  $a$
  - Flakirajuće pridruživanje
  - Provjera je li  $a$  manji ili jednak  $b$
  - Neblokirajuće pridruživanje
5. Memorijski prostor za pohranjivanje objekata u Javi-nom virtualnom stroju naziva se:
- Gomila (engl. *heap*)
  - Javini stogovi (engl. *java stacks*)
  - Virtualna tablica (engl. *virtual table*)
  - Prostor metoda (engl. *method area*)
6. Koja od navedenih formula nije sintaksno ispravna u logici LTL:
- $p \cup (q \cup r)$
  - $GF(p \vee Fq)$
  - $XXGp$
  - $EFp \cup \neg q$
7. U dijelu koda u NuSMV-u:
- ```

next (turn) :=
  case
    turn = myturn & st = c : !turn;
    TRUE : turn;
  esac;

```
- varijabla
- turn*
- ne mijenja svoju vrijednost:
- Ako je zadovoljeno  $turn = myturn \ \& \ st = c$
  - Ona stalno mijenja svoju vrijednost
  - Ona nikad ne mijenja svoju vrijednost
  - Ako nije zadovoljeno  $turn = myturn \ \& \ st = c$
8. U jeziku *Promela* naredba  $a ! b$ :
- Šalje poruku  $b$  preko komunikacijskog kanala  $a$
  - Šalje poruku  $a$  preko komunikacijskog kanala  $b$
  - Definira sinkroni produkt procesa  $a$  i procesa  $b$
  - Definira asinkroni produkt procesa  $a$  i procesa  $b$

9. Zadan je ekspanzirani asinkroni produkt ( $\sigma^w$ ) s vrijednostima varijabli:  
 $x = 2; x = 1; x = 5; x = 4; x = 7; x = 9; x = 10; x = 13;$   
 $x = 12; x = 11;$   
 Odredite za koji od ponuđenih odgovora je LTL formula  $\Box p$  (Gp - "globally" p) istinita

- (a)  $p \equiv (x \geq 10)$
- (b)  $p \equiv (x < 7)$
- (c)  $p \equiv (x < 14)$
- (d)  $p \equiv (x == 7)$

10. Provjera odgovara li specifikacija sklopa (na razini RTL) predloženoj implementaciji sklopa (na razini logičkih vrata) u logičkom smislu, naziva se:

- (a) Provjera logičkog sadržaja
- (b) Provjera modela
- (c) Provjera tvrdnje
- (d) Provjera ekvivalentnosti

11. Prevedite rečenicu prirodnog jezika "Uvijek, ako vrijedi  $\phi$  onda ne vrijedi  $\psi$ ," u formulu vremenske logike LTL:

- (a)  $G(\phi \Rightarrow F \neg \psi)$
- (b)  $G(\phi \Rightarrow \neg \psi)$
- (c)  $\phi \Rightarrow \neg \psi$
- (d)  $\phi \Rightarrow (\phi \cup \neg \psi)$

12. Koja od navedenih formula nije sintaksno ispravna u logici CTL:

- (a)  $EG(req \cup \neg ack)$
- (b)  $EF req \wedge AG ack$
- (c)  $AG A(req \cup ack)$
- (d)  $E(\neg req \cup \neg ack \wedge req)$

13. Koju vrijednost poprima varijabla  $y$  nakon atomice strukture a koju ispisuje `printf` naredba?

```
byte y=3;
...
atomic{ y--; y++; y++; }
printf("y=%d\n", y);
```

- (a)  $y = 2$
- (b)  $y = 4$
- (c)  $y = 0$
- (d)  $y = 3$

14. Monotona funkcija  $F(X) = (\{s0\} \cup X) \cap \{s0, s1\}$  za skup  $S = \{s0, s1, s2\}$ :

- (a) Nema nijednu fiksnu točku
- (b) Ima najmanju i najveću fiksnu točku
- (c) Ima najveću fiksnu točku
- (d) Ima najmanju fiksnu točku

15. Prevedite rečenicu prirodnog jezika: "Ako je na početku vijest neistinita, onda postoji put na kojem će konačno postati istinita," u formulu vremenske logike CTL:

- (a)  $AG(\neg vijest \Rightarrow EF vijest)$
- (b)  $\neg vijest \Rightarrow EF vijest$
- (c)  $\neg vijest \Rightarrow AF vijest$
- (d)  $AG(\neg vijest \Rightarrow AF vijest)$

16. Glavni zadatak formalne specifikacije, kao jedne od tri glavne vrste formalnih metoda, je:

- (a) Prokazati nekonzistentne i dvosmislene specifikacije
- (b) Pripremiti program za provjeru modela
- (c) Omogućiti logičku sintezu programa
- (d) Sastaviti logičke formule za automatizirano dokazivanje teorema

17. Jedan od mogućih ishoda na kraju korištenja postupka "lazy offline" (dodavanjem lema teorije za odlučljivu teoriju) kod SMT-rješavača je:
- SAT i SMT-rješavači kažu da postoji model
  - SAT-rješavač kaže da ne postoji model, a SMT-rješavač kaže da postoji model
  - SAT-rješavač kaže da postoji model, a SMT-rješavač kaže da ne postoji model
  - SMT-rješavač nikada ne završi
18. Složenost izračunavanja algoritma  $ITE(f, g, h)$  ako se ne koristi izračunska tablica, u najgorem slučaju, je:
- Linearna prema broju varijabli funkcija  $f, g$  i  $h$ .
  - Polinomijalna prema broju varijabli funkcija  $f, g$  i  $h$ .
  - Eksponencijalna prema broju varijabli funkcija  $f, g$  i  $h$ .
  - Ovisna o umnošku broja čvorova funkcija  $f, g$  i  $h$ :  $O(|f| \cdot |g| \cdot |h|)$
19. Izrazom `DEFINE turn := myturn;` u NuSMV-u:
- Navodimo ograničenje koje treba uvijek vrijediti za varijablu `turn`
  - Definiramo pravednost koja treba biti ispoštovana
  - Definiramo makro instrukciju
  - Navodimo specifikaciju koju treba provjeriti
20. Zadan je Promela proces `Acounter`:
- ```

1 int i=0;
2 active proctype A1()
3 do
4   :: i==0; i++;
5   :: i<0; goto end_A1;
6 od;
7 end_A1;
8 skip;

```
- Koja je vrijednost varijable `i` na kraju izvođenja procesa. Izvede li se naredba u redu 8 (`skip`)?
- $i = 0$ , `skip` se ne izvede jer dolazi do zastoja
  - $i = 1$ , `skip` se ne izvede jer dolazi do zastoja
  - $i = 0$ , `skip` se izvede i ne dolazi do zastoja
  - $i = 1$ , `skip` se izvede i ne dolazi do zastoja
21. Pri usporedbi logika LTL, CTL i CTL\*, koja od sljedećih izjava vrijedi:
- Logika CTL\* ima manju izražajnost i od logike CTL i od logike LTL
  - Logika LTL ima veću izražajnost i od logike CTL\* i od logike CTL
  - Logika CTL\* ima veću izražajnost i od logike CTL i od logike LTL
  - Logika CTL\* ima različitu i neusporedivu izražajnost u odnosu na logike CTL i LTL
22. Koliko ima istinitih tvrdnji u sljedećem nizu:
- U jeziku *Promela* nema razlike između uvjeta i naredbi, sve su naredbe ili izvršne ili blokirane,
  - Naredba: `atomic(nar1, nar2, nar3)` promatra naredbe `nar1`, `nar2` i `nar3` kao da su nedjeljive tj. izvode se zajedno u bloku,
  - U jeziku *Promela* korisnik može definirati vlastite tipove podataka i podatkovne strukture korištenjem `typedef` naredbe,
  - U sustavu *Spin-Promela* moguće je definirati sustav s najviše 256 aktivna procesa: `active [1024] proctype ProcX() ...`
- četiri
  - tri
  - dvije
  - jedna
23. Glavni problem pri korištenju konkoličkog izvršavanja programa kod programa koji sadrže preteška ograničenja za SMT-rješavač je:
- I simboličko i konkretno stanje programa utječu na izvođenje
  - Operacije se izvedu samo simbolički
  - Nema garancije prolaska svim putovima kroz program
  - SMT-rješavač može zapeti tako da izvođenje programa ne završi
24. Analiza konflikta kod algoritma MiniSAT postiže se rezolucijskim pravilom. Rezolucijom klauzula  $K1 = \neg b \vee \neg c \vee h$  i  $K2 = c \vee e \vee f$  dobiva se klauzula:
- $K3 = \neg b \vee c \vee e \vee f \vee h$
  - $K3 = \neg b \vee e \vee f \vee h$
  - $K3 = b \vee \neg c \vee \neg f \vee \neg h$
  - $K3 = \neg b \vee \neg c \vee e \vee f \vee h$



Student: \_\_\_\_\_

25. Jedna od navedenih formalnih metoda nije metoda formalne specifikacije niti sinteze. Koja?

- (a) Provjera ekvivalentnosti
- (b) TLA i TLA+
- (c) B-metoda
- (d) Z-metoda

26. Ispravno navođenje implicitnog nederminizma u NuSMV-u u stilu dodjeljivanja za varijablu *req* tipa *boolean* je:

- (a) Kod implicitnog nederminizma ne navode se vrijednosti koje varijabla treba poprimiti
- (b) `init(req) := {TRUE, FALSE};`
- (c) `init(req) := {ready, busy};`
- (d) `req = {TRUE, FALSE};`

27. Proširenja Java Pathfindera otkrivaju razna korisnički specificirana svojstva, što je omogućeno mehanizmom:

- (a) Tvrdnji (engl. *assertions*)
- (b) Slušača (engl. *listeners*)
- (c) Ugovora (engl. *contracts*)
- (d) Provjere podudaranja stanja (engl. *state matching*)

28. Ključnom riječju *process* ispred naziva modula u NuSMV-u označavamo da se:

- (a) Modul izvršava sinkrono s drugim takvim modulima, što znači da se njihovi *assign* blokovi izvode istovremeno
- (b) Modul izvodi na asinkroni način, ispreplitanjem izvođenja s drugim asinkronim modulima
- (c) Modul modelira kao Promelin proces, što je nužno za izvođenje Promelinog koda u NuSMV-u
- (d) Modul izvodi na pravedan način, s garancijom da će uvijek biti izabran za izvođenje

29. U formalnom sustavu  $(\Gamma, L)$ , skup formula  $\Gamma$  je \_\_\_\_\_ ako i samo ako ne sadrži formule na temelju kojih bi  $\omega_i$  i  $\neg\omega_i$  istovremeno bile dedukcije.

- (a) Model
- (b) Odlučljiv
- (c) Kompletan
- (d) Konzistentan

30. Za problem raspoređivanja poslova, pri čemu se  $n$  poslova dijeli na  $m$  dijelova koji se raspoređuju na  $m$  računala, uspješno se koristi teorija:

- (a) Rekurzivnih struktura
- (b) Aritmetike razlike
- (c) Peanove aritmetike
- (d) Teorija polja (engl. *arrays*)

31. *Never* blok (`never( . . . . . )`) u jeziku *Promela*:

- (a) Nakon naredbe *timeout* definira protuprimjere ako dođe do zastoja (engl. *deadlock*)
- (b) Realizira iznimke (engl. *exceptions*) u jeziku *Promela*
- (c) Definira Kripke strukturu u jeziku *Promela*
- (d) Je formula LTL logike implementirana kao Büchi automat u sintaksi jezika *Promela*

32. Booleova funkcija  $F = a \cdot \bar{b}$  može se prikazati u obliku operatora ITE kao:

- (a)  $ITE(a, \bar{b}, 1)$
- (b)  $ITE(a, \bar{b}, 0)$
- (c)  $ITE(a, 0, \bar{b})$
- (d)  $ITE(a, 1, \bar{b})$

Student: \_\_\_\_\_

33. Korolar teorema o dedukciji kaže da je formula  $\psi$  logička posljedica formule  $\phi$ , tj.  $\phi \models \psi$ , ako i samo ako je formula:

- (a)  $(\phi \wedge \neg\psi)$  nezadovoljiva
- (b)  $(\phi \wedge \neg\psi)$  tautologija
- (c)  $(\phi \vee \neg\psi)$  tautologija
- (d)  $(\phi \vee \neg\psi)$  nezadovoljiva

34. Zadan je dio *Promela* procesa:

```
1 int n=0;
2 do
3   :: n == 0; n++;
4   :: n == 1; n=n-2;
5   :: n == 2; n++;
6   :: n < 0 -> timeout;
7 od
8 skip;
```

Odredite vrijednost varijable  $n$  u redu 8.

- (a)  $n = 1$
- (b)  $n = 2$
- (c)  $n = 3$
- (d) dolazi do globalnog zastoja (*deadlock-timeout*), naredba *skip* u redu 8 se ne izvede,  $n = -1$

Ljetni ispitni ispit – 2. dio

(maks. 36 bodova)

IME I PREZIME: \_\_\_\_\_ JMBAG: \_\_\_\_\_

1. (4 boda) Dokažite istinitost sljedećeg zaključka koristeći pravila prirodnog zaključivanja L koja vrijede u propozicijskoj logici:

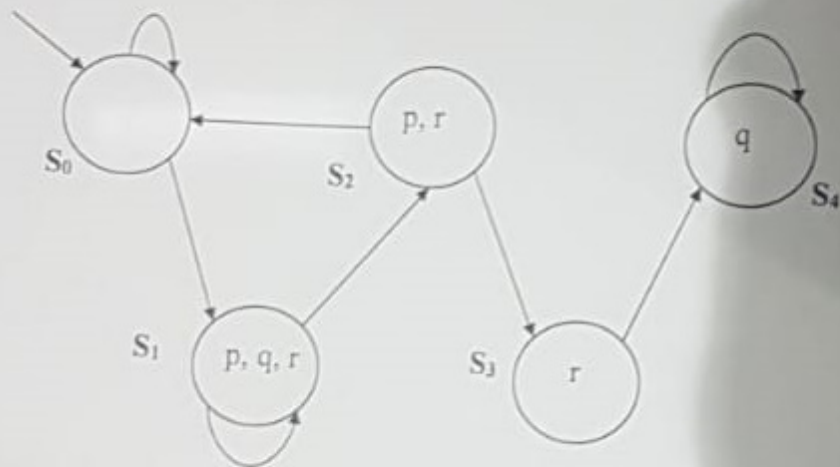
$$(P \Rightarrow Q) \vdash \neg ((P \wedge Q) \Rightarrow P) \wedge (P \Rightarrow (P \wedge Q))$$

Napomena: u dokazu trebate u svakom koraku sa strane navesti radi li se o premisi, pretpostavci ili pravilu, a ako se koristi pravilo onda trebate navesti nad kojim prethodnim koracima je pravilo primijenjeno.

2. (2 boda) Definirajte potrebne predikate i konstante i zatim preslikajte rečenicu prirodnog jezika u dobro definiranu formulu predikatne logike prvoga reda (FOPL):

"Svaki računalni sustav sadržava više od jednog programa."

3. (3 boda) Koristeći teoriju fiksne točke i odgovarajući algoritam, odredite  $Q(EG\ q)$  za Kripkeovu strukturu prikazanu na slici (potrebno je napisati cjelokupni postupak dobivanja rješenja i konačno rješenje!).



4. (3 boda) Za programski kôd zadan u programskom jeziku Verilog:  
 a) (2 boda) nacrtajte odgovarajući stroj s konačnim brojem stanja (FSM);  
 b) (1 bod) navedite o kojoj vrsti stroja s konačnim brojem stanja se ovdje radi.

```

typedef enum {S1,S2,S3,S4} state;

module kripke(clk, x, y);
input clk;
state reg s;
output x,y;
wire x,y,rand;

assign rand = $ND(0,1);
assign x = (s == S1) ? (rand == S3);
assign y = (s != S4);

initial begin
    s = S1;
end

always @(posedge clk) begin
    case(s)
        S1: if (rand) s = S2;
        S2: if (rand) s = S1;
            else s = S3;
        S3: if (rand) s = S4;
        S4: if (rand) s = S1;
    endcase
end
endmodule

```

5. (3 boda) Navedite i ukratko objasnite na koja tri načina Java Pathfinder nastoji doskočiti problemu eksplozije broja stanja koja se događa u Javinim programima.

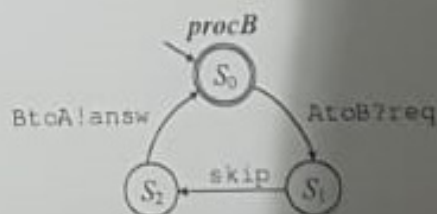
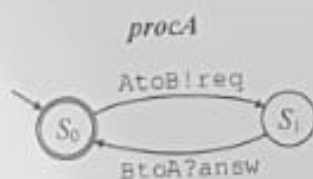
6. (2 boda) Zaokružite jesu li sljedeće tvrdnje istinite (T) ili neistinite (N):

- a) T N U jeziku *Promela* se naredbom `break` prekida do petlja.
- b) T N Büchi automat prihvata konačne sekvence ( $\sigma^w$ ) labela  $L$ .
- c) T N U jeziku *Promela* naredba `a!b` šalje poruku  $b$  u komunikacijski kanal  $a$ .
- d) T N Naredbom `spin -f "[<>p]"` se *LTL* formula prevodi u `never` blok u jeziku *Promela*.

Obrazložite svaki od odgovora!

- a) \_\_\_\_\_
- b) \_\_\_\_\_
- c) \_\_\_\_\_
- d) \_\_\_\_\_

7. (3 boda) Na slici su prikazana dva konačna automata (FSA): *procA* i *procB*.



Za zadane automate:

- a) Napisati pripadne naredbe za *Promela* procese ako *procA* i *procB* izmjenjuju poruke preko sinkronih kanala `AtoB` i `BtoA`, tako da nadopunite sljedeći predložak:

```

active proctype procA() {
    active proctype procB() {

```



- b) Odredite (nacrtajte) asinkroni produkt automata *procA* i *procB*:  
 $C_{FSA} = procA \times procB = (C.S, C.s_0, C.L, C.T, C.F)$

- c) Ako umjesto sinkronih kanala uvedemo asinkrone kanale kapaciteta 5, obrazložite koje je naredbe potrebno modificirati. Objasnite da li je u tom slučaju moguć zastoј (eng. *deadlock*).

---



---



---

8. (2 boda) Zadani su sljedeći *Promela* procesi *ProcA* i *ProcB*:

int x = 2;

```
active proctype ProcA() {
  if
    :: x < 10 -> x++;
    :: else -> x--;
  fi;
}
```

```
active proctype ProcB() {
  do
    :: x <= 3 -> x=x*5;
    :: x > 10 -> x--;
    :: else -> break;
  od;
}
```

- a) Nacrtajte konačne automate za *Promela* procese *ProcA* i *ProcB*.

- b) Opisati postupak kako primjenom *LTL* formule odrediti istinitost sljedeće tvrdnje te postupak pronalaženja protuprimjera (eng. *counterexample*):  
 „Tijekom izvođenja globalna varijabla *x* neće poprimiti vrijednost  $x > 15$ .“

---



---



---

c) Koju vrijednost poprima globalna varijabla  $x$ ? Objasnite rješenje!

---

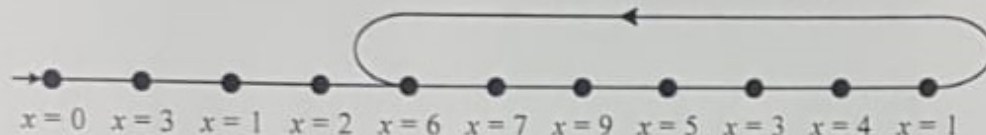


---



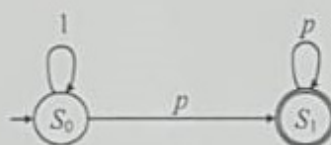
---

9. (2 boda) Na slici je prikazan ekspanzirani asinkroni produkt ( $\sigma^w$ ) s vrijednostima varijabli. Odredite (zaokružite odgovor) da li su *LTL* formule istinite (T) ili neistinite (N):



- |   |   |  |
|---|---|--|
| T | N | $\Box \Diamond p$ ako je $p \equiv (x = 5)$    |
| T | N | $\Diamond p$ ako je $p \equiv (x > 10)$        |
| T | N | $\Diamond \Box p$ ako je $p \equiv (x \geq 1)$ |
| T | N | $\Box p$ ako je $p \equiv (x = 2)$             |

10. (1 bod) Na slici je prikazan Büchi automat.



a) Napišite temporalnu *LTL* formulu koju takav automat realizira:

b) Nadopunite `never { }` tvrdnju s *Promela* instrukcijama koje realiziraju taj automat:

`never {`

`}`

11. (4 boda) Za funkciju bita prijenosa  $C_{out}$  (2-bitnog) potpunog zbrajala zadanu tablično:

- (1 bod) nacrtajte BDD,
- (2 boda) BDD svedite na ROBDD,
- (1 bod) dodajte komplementirane lukove ROBDD-u te nacrtajte konačni ROBDD.

**Napomene:**

- Pretpostavite da je uređenje  $A < B < C_{in}$ .
- Svođenje BDD-a na ROBDD i ROBDD-a na ROBDD s komplementiranim lukovima treba biti prikazano iz koraka u korak.
- Zadatak trebate riješiti na način da ne računate Shannonovu ekspanziju, već korištenjem postupka za svođenje BDD-a na ROBDD.
- Zadatak riješite u košuljicu ili na dodatni papir.

A	B	C <sub>in</sub>	C <sub>out</sub>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

12. (4 boda) Za zadanu bazu klauzula provedite algoritam GRASP i pronađite odgovor na pitanje je li baza klauzula zadovoljiva (SAT) ili ne. Ako je SAT, napišite konačno rješenje. Pri rješavanju, crtajte graf implikacija (na određenoj razini odluke) samo onda kada postoje implikacije (forsirane vrijednosti literala). Ako dođe do konflikta tijekom implikacija, dodajte novonaučenu klauzulu i odredite razinu odluke na koju se postupak vraća te nastavite s postupkom. Kao heuristiku pri grananju koristite onu koja gleda koji je literal najčešći u nekom trenutku i koristi taj najčešći literal za smanjenje veličine klauzule (ne za eliminaciju klauzula). Ako ima više najčešćih literala u nekom trenutku, odaberite jedan po volji. **Napomena:** zadatak riješite u košuljicu ili na dodatni papir.

$$K1: x1 + \neg x2 + x5 + x4$$

$$K2: x1 + x2 + \neg x5 + \neg x4$$

$$K3: \neg x3 + x6$$

$$K4: x4 + x7 + x1$$

$$K5: \neg x4 + \neg x7 + x1$$

13. (3 boda) Za SAT-rješavače:
- (1 bod) Objasnite zašto se uvodi brisanje naučenih klauzula kod učinkovitih suvremenih algoritama za rješavanje SAT-problema.
  - (2 boda) Navedite i objasnite strategiju koju koristi SAT-rješavač Chaff za brisanje naučenih klauzula.