ZADACI ZA VJEŽBU (MI)

1. LOGIKA

Propozicijska logika

1. Pretvorite prop. formulu u CNF-oblik

$(P \Rightarrow (Q \Rightarrow R)) \Leftrightarrow (P \Rightarrow (R \Rightarrow Q))$

1 se rješavamo ekvivalencije tako da je pretvorimo u impl.

$(P \Rightarrow (Q \Rightarrow R)) \Rightarrow (P \Rightarrow (R \Rightarrow Q)) \wedge (P \Rightarrow (R \Rightarrow Q) \Rightarrow (P \Rightarrow (Q \Rightarrow R)))$

$\neg(\neg P \vee \neg Q \vee R) \vee (\neg P \vee (\neg R \vee Q)) \wedge \neg(\neg P \vee (\neg R \vee Q) \vee (\neg P \vee \neg Q \vee R))$

$(P \wedge Q \wedge \neg R) \vee (\neg P \vee \neg R \vee Q) \wedge (P \wedge R \wedge \neg Q) \vee (\neg P \vee \neg Q \vee R)$

$Q \wedge \underbrace{(P \wedge \neg R \vee 1)}_{=1} \vee (\neg P \vee \neg R) \wedge (\neg Q \wedge \underbrace{(P \wedge R \wedge 1)}_{=1} \vee \neg P \vee R$

$(Q \vee \neg P \vee \neg R) \wedge (\neg Q \vee \neg P \vee R)$

2. Koristeći teorem o dedukciji, ekvivalencije koje vrijede u propozicijskoj logici i pravila zaključivanja, pokažite da je formula S logička posljedica skupa formula:

1. P
2. $(P \Rightarrow Q)$
3. $(Q \Rightarrow S)$

Rj:

$\underline{\varphi \Rightarrow \psi}$ $\underline{\varphi \wedge \neg\psi}$

P

$P \Rightarrow Q$

$Q \Rightarrow S$

$\neg S$

vrijedi:

1. P, $P \Rightarrow Q$ modus ponens (pa vrijedi Q)

Q

2. $Q \Rightarrow S$, Q Modus ponens (pa vrijedi S)

S

$\neg S$ 3. $S, \neg S$ – kontradikcija – znači da je logička posljedica

3. Koristeći pojam logičke posljedice, pokažite ili opovrgnite semantičku ekvivalentnost formula:

$\varphi_1 : P \Rightarrow (Q \vee R)$;

$\varphi_2 : (\neg Q \wedge \neg R) \Rightarrow P$

Formule su ekvival. ako vrijedi $\varphi_1 \models \varphi_2 \wedge \varphi_2 \models \varphi_1$

P	Q	R	$\neg Q$	$\neg R$	$Q \vee R$	$\neg Q \wedge \neg R$	φ_1	φ_2
0	0	0	1	1	0	1	1	0
0	0	1	1	0	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	0	0	1	0	1	1
1	0	0	1	1	0	1	0	1
1	0	1	1	0	1	0	1	1
1	1	0	0	1	1	0	1	1
1	1	1	0	0	1	0	1	1

nema logičke posljedice

PREDIKATNA LOG.

4. Preslikajte slj. rečenice prir. jezika u formalizam predik. logike prvog reda (FOPL). Pritom definirajte sve potrebne predikate i konstante

a) "Svaki profesor je zaposlenik samo jednog fakulteta, a predaje na jednom ili više fakulteta."

PROF(x) : x je profesor

ZAPOSLEN_U(x,y) : x zaposlen u y

PREDAJE_U(x,y) : x predaje u y

FAKULTET(x) : x je fakultet

$$\forall x\ (PROF(x)) \Rightarrow \exists y\ (FAKULTET(y) \wedge ZAPOSLEN\_U(x,y) \wedge \neg \exists z\ (FAKULTET(z) \wedge ZAPOSLEN\_U(x,z) \wedge \neg = (y,z))) \wedge \exists y\ (PREDAJE\_U(x,y) \wedge FAKULTET(x,y))$$

b) "U nekom programu postoje definirana najmanje dva različita razreda."

PROGRAM(x) : x je program

RAZRED(x) : x je razred

DEFINIRAN_U(x, y) : x je definiran u y

$\exists x (\text{PROGRAM}(x) \wedge \exists y \ \text{RAZRED}(y) \wedge \exists z (\text{RAZRED}(z) \wedge \text{DEFINIRAN\_U}(y, x) \wedge$
$\text{DEFINIRAN\_U}(z, x) \wedge \neg = (y, z)))$

c) "Svatko voli nekog i nitko ne voli svakog."

VOLI(x, y) : x voli y

$\forall x \exists y (\text{VOLI}(x, y)) \wedge \neg \exists x \forall y (\text{VOLI}(x, y))$

CTL LOGIKA

5. Koje su od navedenih sintaksno dobro definirane formule u logici CTL:

a) $p \wedge \neg q \vee EX\, p$

~~b)~~ $AF(G\, p \Rightarrow GF\, q)$ — može biti u CTL*

c) $AG\ AF\ EX\ (p \wedge q)$

~~d)~~ $A((p\, U\, q) \vee (q\, U\, \neg r))$ $A(p U q) \vee A(q U \neg r)$

~~e)~~ $E(p \Rightarrow E(q \Rightarrow r))$

f) $A((q \wedge r)\, U\, (p \wedge r))$

6. Preslikajte rečenice prir. jezika u formule logike CTL.

a) "Uvijek u svim stanjima sustava vrijedi da p ne vrijedi dok ne počne vrij. q."

$AG(A \neg p \, U \, q)$

b) "Postoji put u kojem se konačno dolazi do stanja u kojem vrijedi p i od kojeg dalje p ne vrijedi u sljedeća dva stanja."

$EF(p \wedge AX \neg p \wedge AXAX \neg p) \wedge AX(\neg p \wedge AX \neg p)$

c) "U početnom stanju vrijedi p, a zatim u sljedećem stanju može ne vrijediti q."

$\cancel{p \wedge EX \neg q}$ / $p \wedge AX \underbrace{\neg q \vee q}_{TRUE} \rightarrow p \wedge AX \, TRUE$

LTL LOGIKA

7. Preslikajte rečenice prir. jezika u formule LTL:

a) "Nije moguće doći u stanje gdje vrijedi p i ne vrijedi q."

$\neg F(p \wedge \neg q)$

$G!(p \wedge \neg q)$

b) "Uvijek se konačno dolazi u stanje gdje vrijedi p, a od idućeg stanja p vrijedi dalje beskonačno često."

$F(p \wedge XGFp)$

c) "Uvijek ako vrijedi p, q će vrijediti od tog stanja sve dok p ne prestane vrijediti."

$G(p \Rightarrow q \, U \, \neg p)$

FIKSNA TOČKA I EKSPLIC. IZRAČ. STANJA

8. Za f-ju $F(x) = (x \cup [s1]) \cap [s2]$ i skup mogućih stanja $S = [s0, s1, s2]$ odredite:

a) Je li f-ja monotona

b) Ako f-ja jest monotona, nađite najmanju i najveću fiksnu točku

$P(S) = \{\}, \{s0\}, \{s_1\}, \{s2\}, \{s0, s1\}, \{s0, s2\}, \{s1, s2\}, \{s0, s1, s2\}$

$X_1 \subseteq X_2 \Rightarrow F(X_1) \subseteq F(X_2)$

Rj:

a) Uvrštavati:

$\{\} \subseteq \{s0\} \Rightarrow F(\{\}) = \{\}$ ✓

$F(\{s_0\}) = \{\}$

$(\{\} \cup \{s_1\})$

$\{s_1\} \underset{\text{presjek}}{\cap} \{s_2\} = \{\}$

$\{\} \subseteq \{s_0, s_1, s_2\} \Rightarrow F(\{\}) = \{\} \subseteq F(\{s_0, s_1, s_2\}) = \{s_2\}$

$\{s_0\} \subseteq \{s_0, s_2\} \Rightarrow F(\{s_0\}) = \{\} \subseteq F(\{s_0, s_2\}) = \{s_2\}$

MAX. skup koji možemo dobiti je $\{s_2\}$ jer je presjek sa $\{s_2\}$ u zadatku.

$\{s_0, s_1\} \subseteq \{s_0, s_1, s_2\} \Rightarrow F(\{s_0, s_1\}) = \{\}$

$F(S) = \{s_2\}$

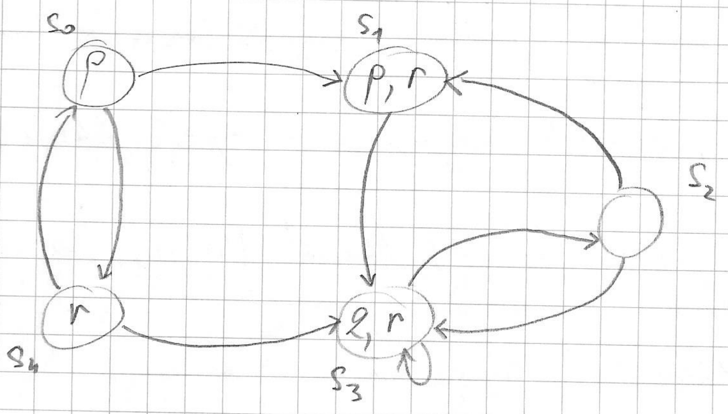
F-ja je monotona

b) Primijeniti f-ju: (na onoliko koliko u skupu ima elemenata)

$F^3(\{\}) = F(F(F(\{\}))) = F(F(\{\})) = F(\{\}) = \{\}$ $\emptyset$

$F^3(\{s_0, s_1, s_2\}) = F(F(F(\{s_0, s_1, s_2\}))) = F(F(\{s_2\})) = F(\{s_2\}) = \{s_2\}$

9. Koristeći algoritam fiksne točke, izračunajte skup stanja koja zadovoljavaju CTL formulu E p U r za zadanu Kripkeovu strukturu.



$Q(E\ p\ U\ r) \qquad Z_0 = \emptyset$

$Z_{k+1} = Q(r) \cup (Q(p) \cap R^{-1}(Z_k))$

$Z_1 = \{1,3,4\} \cup (\{0,1\} \cap \emptyset) = \{1,3,4\} \neq Z_0$

$Z_2 = \{1,3,4\} \cup (\{0,1\} \cap \{\underbrace{0,2}_{\text{inv. za } 1}, \underbrace{1,3,4}_{\text{inv. za } 3} \}) = \{0,1,3,4\} \neq Z_1$

inv. za 4 je 0

inverzna od $\{1,3,4\}$

$Z_3 = \{1,3,4\} \cup (\{0,1\} \cap \{4,0,2,1,3\} = \{0,1,3,4\} = Z_2$

Sustav NuSMV

10. Za zadani kod modula u jeziku NuSMV potrebno je nacrtati odgovarajuću Kripkeovu strukturu i potom odrediti istinitost dvije CTL specifik., uzevši u obzir navedeno ograničenje pravednosti:

a) CTLSPEC AF EG material = present

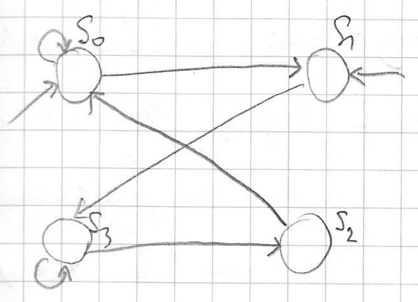
b) CTLSPEC AG (line_on → EX (AX ! line_on))

MODULE factory

VAR

line_on: boolean;

material: {absent, present};



$S_0 = \{ !\text{line\_on}, \text{absent} \}$

$S_1 = \{ !\text{line\_on}, \text{present} \}$

$S_2 = \{ \text{line\_on}, \text{absent} \}$

$S_3 = \{ \text{line\_on}, \text{present} \}$

```
ASSIGN
  init (lin_on) := FALSE;
  next(line_on) := case
                   (material = present) : TRUE;
                   TRUE : FALSE;
                   esac;
  next(material) := case
                   (line_on & material = absent) : absent;
                   (!line_on & material = present) : present;
                   TRUE : (absent, present);
                   esac;
DEFINE output = line_on & material = present;
JUSTICE material != absent;
```

a) AF EG material=present → vrijedi za S_3 m = present
↓
TRUE

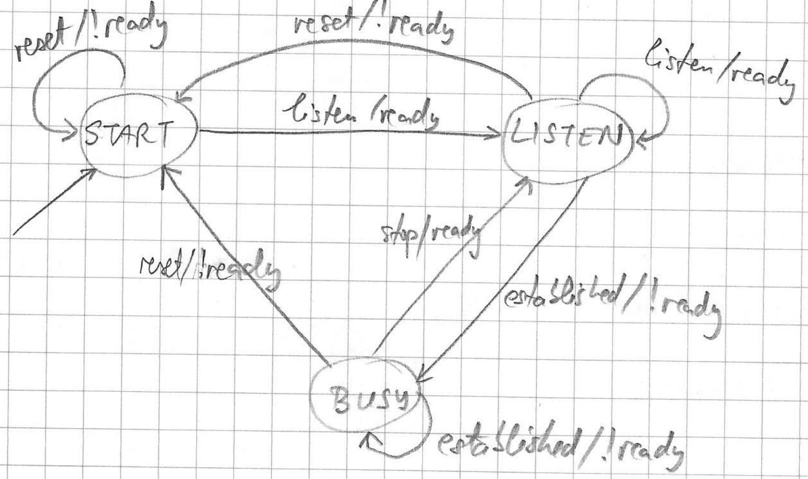
a u S_0 se može ostati beskonačno dugo
ALI JUSTICE ne dopušta da material bude absent
beskonačno često, tako da ni u stanju S_0
(bez JUSTICE ne bi vrijedila formula

b) AG (line_on → EX (AX !line_on)) TRUE

da je bilo AX (AX !line_on) – ne bi vrijedilo

VERILOG + VIS

11. Za zadani stroj s konačnim brojem stanja (dalje: FSM) potrebno je napisati odgovarajući kod modula server_connection u jeziku Verilog. Pripaziti na vrste FSM-a. Pretpostavite da se promjene stanja događaju na pozitivan brid eksplicitno zadanog ulaznog signala takta. Ulazi i izlazi modula trebaju odgovarati onima na zadanome FSM-u.



Milijev

```
TYPEDEF ENUM {START, LISTEN, BUSY} STATE;
MODULE SERVER_CONNECTION (CLK, LISTEN, READY, ESTABLISHED, STOP, RESET);
INPUT CLK, RESET, LISTEN, ESTABLISHED, STOP;
OUTPUT READY;
STATE REG ST;
WIRE RESET, LISTEN, ESTABLISHED, STOP, READY;
ASSIGN READY = ((ST==START && LISTEN) || (ST==LISTEN && LISTEN) || (ST==BUSY
    && STOP));
INITIAL ST=START;
ALWAYS @(POSEDGE CLK) BEGIN
    CASE (ST)                                   ASSIGN
      START: IF (RESET) ST=START;                 READY=(STATE==BUSY);
             ELSE IF (LISTEN) ST=LISTEN;
      LISTEN: IF (RESET) ST=START;
             ELSE IF (LISTEN) ST=LISTEN;
             ELSE IF (ESTABLISHED) ST=BUSY;
```

```
        BUSY: IF (RESET) ST=START;
              ELSE IF (STOP) ST=LISTEN;
              ELSE IF (ESTABLISHED) ST=BUSY;
    END CASE
  END
END MODULE
```

Java Path Finder

- konfiguriranje datoteka - gdje se nalaze koje datoteke,.. koji im je sadržaj
  kako dodati novog listenera