MI ZADACI ZA VJEŽBU 2014/2015

(3.) PREDIKATNA LOGIKA -> upsić, poćela sam s trećim 😉

SVAKI PROFESOR JE ZAPOSLENIK TOČNO JEDNOG FAKULTETA, A PREDAJE NA JEDNOM ILI VIŠE.

PROF(x) - x je prof.

F(x) - x je fakultet

Z(x,y) - x je zaposlen na y

PREDAJE(x,y) - x predaje na y

=(x,y) - x je jednako y

$$\forall x [PROF(x) \Rightarrow \underbrace{\exists y [F(y) \wedge Z(x,y) \wedge \neg \exists z (F(z) \wedge Z(x,z) \wedge \neg =(y,z))]}_{\text{zaposlenje}} \wedge \underbrace{\exists y [F(y) \wedge PREDAJE(x,y)]}_{\text{predavanja}}$$

prijevod: svaki profesor x povlači da postoji y takav da je y fakultet i profesor x je zaposlen na fakultetu y i ne postoji z takav da je z fakultet i profesor x je zaposlen na fakultetu z i fakultet z je različit od fakulteta y. I da postoji y takav da je fakultet y i profesor x predaje na fakultetu y

* primjetite da tu nema uvjeta $y \neq z$

U NEKOM PROGRAMU DEFINIRANA SU NAJMANJE DVA RAZLICITA RAZREDA

$P(x)$ - x je program

$R(x)$ - x je razred

$DEF(x,y)$ - x je definiran u y

$=(x,y)$ - x je jednak y

$$\exists x \exists y \exists z \left[P(x) \wedge R(y) \wedge R(z) \wedge DEF(y,x) \wedge DEF(z,x) \wedge \neg =(y,z) \right]$$

SVATKO VOLI ~~NEKOG~~ ~~SVAKOG~~ I NITKO NE VOLI SVAKOG

$VOLI(x,y)$ - x voli y

$$\forall x \forall y [VOLI(x,y)] \wedge \neg \exists x \forall y [VOLI(x,y)]$$

(1.) PROPOZICIJSKA LOGIKA

PRETVORI U CNF (konjunkcijski normalni oblik) → sve mora biti povezano s V unutar zagrade i ∧ izvan zagrade

$(A \Rightarrow B) = (\neg A \vee B)$

$(A \Leftrightarrow B) = ((A \Rightarrow B) \wedge (B \Rightarrow A))$

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

$$\equiv (P \Rightarrow (\neg Q \vee R)) \Rightarrow (P \Rightarrow (\neg R \vee Q))$$

$$\equiv (\neg P \vee (\neg Q \vee R)) \Rightarrow (\neg P \Rightarrow (\neg R \vee Q))$$

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

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

$$1 \equiv (Q \vee \neg P \vee \neg R) \wedge (\neg R \vee \neg P \vee Q)$$

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

(2)

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

$$\equiv P \wedge (\neg P \vee Q) \wedge (\neg Q \vee S) \wedge (\neg S)$$

$$\equiv [\underbrace{(P \wedge \neg P)}_{0} \vee (P \wedge Q)] \wedge [(\neg Q \wedge \neg S) \vee \underbrace{(S \wedge \neg S)}_{0}]$$

$$\equiv (P \wedge Q) \wedge (\neg Q \wedge \neg S)$$

$$\equiv (P \wedge \underbrace{Q \wedge \neg Q}_{0} \wedge \neg S)$$

$0 \Rightarrow$ sve je $0 \Rightarrow S$ je logička posljedica

(4.) CTL Logika

kvantifikator { E - postoji put; A - za svaki put }

a) $P \wedge \neg Q \vee EX_P$

- ispravno

operator stanja { X - u slijedećem stanju; G - za sva stanja; F - postoji stanje; U - od ovog stanja do nekog stanja }

b) $AF(GP \Rightarrow GFQ)$

- neispravno, operator stanja UVIJEK dolazi u paru s kvantifikatorom

c) $AGAFEX(P \wedge Q)$

- ispravno

d) $A((PUQ) \vee (QU \neg R))$

- neispravno, za CTL uz svaki U mora stajati kvantifikator, dok bi u CTL* ovo vrijedilo

e) $E[P \Rightarrow E(Q \Rightarrow R)]$

- neispravno, kvantifikatori nemaju operator stanja uz sebe

f) $A((Q \wedge R)U(P \wedge R))$

- ispravno, jedan operator stanja U ima svoj kvantifikator

(5.)

a) UVIJEK U SVIM STANJIMA SUSTAVA VRIJEDI DA P NE VRIJEDI DOK NE POČNE VRIJEDITI Q

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

b) POSTOJI PUT U KOJEM SE KONAČNO (postoji stanje) DOLAZI DO STANJA U KOJEM VRIJEDI p I OD KOJEG DALJE p NE VRIJEDI U SLIJEDEĆA 2 STANJA

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

↳ to je za prvo slijedeće stanje ↳ to je za drugo stanje

c) U POČETNOM STANJU VRIJEDI p, A ZATIM POSTOJI PUT NA KOJEM U SLIJEDEĆEM STANJU NE VRIJEDI Q

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

↳ početno stanje

6. LTL logika

a) NIJE MOGUĆE DOĆI U STANJE GDJE VRIJEDI p I NE VRIJEDI Q (nikad)

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

b) UVIJEK SE KONAČNO DOLAZI U STANJE GDJE VRIJEDI p, A NAKON TOG STANJA p VRIJEDI BESKONAČNO ČESTO

$$F(p \wedge GF(p))$$

c) UVIJEK AKO VRIJEDI p, q ĆE VRIJEDITI OD TOG STANJA DOK p NE PRESTANE VRIJEDITI

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

; LTL ima samo vremenske operatore
; F - konačno
; G - uvijek
; X - u slijedećem koraku
; U - p dok ne počne vrijediti q
; u LTL-u kvantifikator A je implicitan (podrazumijeva se), a kvantifikator E ne postoji

⑦ FIKSNA TOČKA I BESKONAČNO IZRAČUNAVANJE STANJA

EpUr — FIXNA točka – za ta stanja vrijedi cudumi uvijet

$z_0 = \emptyset$ – uzimamo uvijek kada imamo $\cup$, da smo imali EG u skupu bi bili svi

↗ skup stanja kada vrijedi r

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

↳ iz kojih stanja možemo doći u to stanje

za EG se razlikuje!

$z_{k+1} = Q(r) \cap R^{-1}(z_k)$

$Q(r) = \{1,3,4\}$

$Q(p) = \{0,1\}$

$R^{-1}(z_0) = 0$

$z_1 = \{1,3,4\} \cup (\{0,1\} \cap \emptyset) = \{1,3,4\}$

$R^{-1}(z_1) = \{0,2,1,4,3\}$

↳ gledam po slici

$z_2 = \{1,3,4\} \cup (\{0,1\} \cap \{0,2,1,4,3\}) = \{1,3,4\} \cup \{0,1\} = \{0,1,3,4\}$

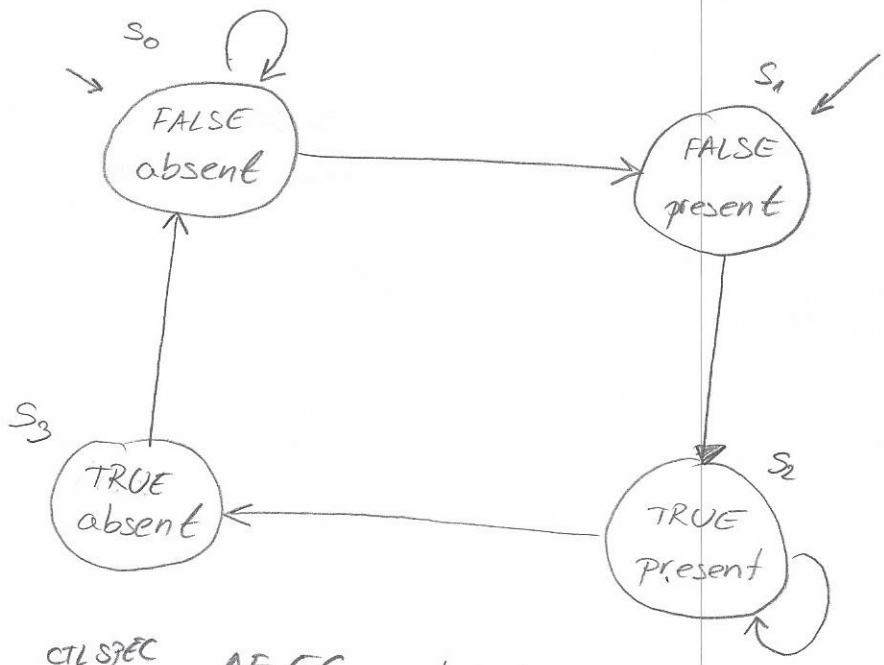
$z_1 \neq z_2$ – nismo došli do fixne točke

$R^{-1}(z_2) = \{0,1,2,3,4\}$

$z_3 = \{0,1,3,4\} \cup (\{0,1\} \cap \{0,1,2,3,4\} = \{0,1,3,4\} \rightarrow \underbrace{z_2 = z_3}$

DOŠLI SMO DO FIXNE TOČKE

(8.) Sustav NuSMV



a) CTLSPEC AF EG material = present

Za svaki put postoji stanje iz kojeg postoji put gdje za sva stanja vrijedi material = present

Vrijedi - iz S_1 dođemo u S_2 gdje možemo zapeti u ujećnoj petlji, dakle postoji put - Da, S_2

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

(9.)

1) NuSMV -int
2) read_model -i ex1.smv
3) go
4) print_reachable_states
5) check_spec -n 1

(10.)

```
typedef enum {START, LISTEN, BUSY} conn_state;   // stanja

module server_connection(clk, listen, established, ready, reset, stop);   // svi inputi i outputi
input clk;
input listen;
input established;
input stop, reset;
output ready;

conn_state reg state;
wire listen, established, stop, ready, reset;
assign ready = [(state = START && listen) ||
                (state = LISTEN && listen) ||
                (state = BUSY && stop)];
initial state = START;
always @(posedge clk) begin   // na pozitivan brid
    if (reset) state = START;
    else if (listen) state = LISTEN;
    else if (established) state = BUSY;
    else if (stop) state = LISTEN;
end
endmodule
```

OVO SE MEALY MODEL, na crti je input/output da je bio MOORE model na crti bi bio input, a output bi pisao u stanju.

← assign ajjek ide na outpute kao kombinacija stanja i inputa

* moglo biti napisano

```
case(state)
    START: if (listen) state = LISTEN;
           else if (reset) state = START;
    LISTEN: if (reset) state = START;
            else if (listen) state = LISTEN;
            else if (established) state = BUSY
    BUSY: if (stop) state = LISTEN;
          else if (reset) state = START;
          else if (established) state = BUSY;
end case
```

(11)

1) read_bliff_mv server_connection.mv
2) print_models
3) init_verify
4) read_fairness server_connection fair
   module_check -i server_connection.ctl

MI 2012/2013

(1.) U formalnim metodama usredotočenost je na skatičko (simboličko) rasuđivanje o sustavima.

(2.) ~~Klasifikacija~~ Klasifikacija formalnih metoda:

1) FORMALNA SPECIFIKACIJA (ASM)
2) FORMALNA SINTEZA (SDL)
3) FORMALNA VERIFIKACIJA (SAT)

(3.) 2 metoda formalne specifikacije

Sheme
1) State schema - globalne izjave o sustavu
2) Operation schema - opisuje učinak određenih operacija koje mijenjaju stanje podataka u sustavu
3) Observation schema - opisuje dohvat informacija, podaci u sustavu se ne mijenjaju

Logika : koristi logiku predikata

(4.) KADAV ~~FM~~ JE ISPRAVAN I KOMPLETAN

Formalan sustav $\{\Gamma, L\}$ je ISPRAVAN (sound) ako je svaki teorem logička posljedica skupa formula $\Gamma \vdash_L \omega_i \Rightarrow \Gamma \models \omega$

Formalan sustav $\{\Gamma, L\}$ je KOMPLETAN (complete) ako je svaku logičku posljedicu skupa moguće dokazati pravilima L tj. $\Gamma \models \omega_i \Rightarrow \Gamma \vdash_L \omega_i$

(5.)
a) $A \Rightarrow B \Rightarrow C$ koristimo lijevu asocijativnost

$A \Rightarrow (B \Rightarrow C) \equiv \neg A \vee (B \Rightarrow C) = \neg A \vee \neg B \vee C$ → to je i CNF (sjecište članom) i DNF (s tri člana)

b) $\neg P \Rightarrow (P \Rightarrow Q) \equiv P \vee (P \Rightarrow Q) \equiv \underbrace{P \vee \neg P}_{1} \vee Q \equiv 1$

→ U ISPRAVNOM i KOMPLETNOM sustavu vrijedi

$\Gamma \models \omega_i = \Gamma \vdash_L \omega_i$ (logička posljedica je ujedno teorem i obratno)

(6.)

a) $\neg(P(c) \vee Q(c)) = \neg P(c) \wedge \neg Q(c)$ ✓ - negacija može u zagradu

b) $\exists x\,(P(x) \Rightarrow Q(x)) = \exists x\,(\neg P(x)) \vee \exists x\,(Q(x))$ ✓ - egzistencijski kvantifikator može ući u zagradu ako je disjunkcija (V) da je bila konjunkcija $\exists x$ nebi mogao u zagradu, ali bi mogao $\forall x$ (univerzalni kvant.)

c) $\forall x(P(x) \Rightarrow Q(x) \Rightarrow R(x)) = \forall x((P(x) \wedge Q(x)) \Rightarrow R(x))$.

↑ univo prethodna implikacija trebalo bi biti $\neg P(x) \vee Q(x)$

(7.)

a) ANTE IMA BAREM DVIJE SESTRE

sestra(x) - x je Antina sestra

$=(x, y)$ - x je jednak y

$\exists x \exists y [\text{sestra}(x) \wedge \text{sestra}(y) \wedge \neg =(x, y)]$

b) ZA SVAKI BRIJEG U HRVATSKOJ POSTOJI VIŠI BRIJEG U HERCEGOVINI

HR(x) - x je brijeg u Hrvatskoj

HERC(x) - x je brijeg u Hercegovini

VISI(x, y) - x je viši od y

$\forall x \exists y [HR(x) \wedge HERC(y) \wedge VISI(y, x)]$ - zašto ovo ne valja?? implikacija povlači da y nužno postoji?

$\forall x [HR(x) \Rightarrow \exists y [HERC(y) \wedge VISI(y, x)]]$

(8) a) AF preko EG

$AF p \equiv \neg EG(\neg p)$

b) AG preko EF

$AG p \equiv \neg EF(\neg p)$

c) AF preko AU

$AF p \equiv A(TRUE\ U\ p)$

d) EG preko AU

$EG p \equiv \neg A(false\ U \neg p)$

- A - svi putevi
- E - postoji put
- F - postoji stanje
- G - sva stanja
- X - slijedeće stanje
- U - vrijedi dok <uvjet>

PREMA WIKIPEDIJI

$EF p \equiv E[TRUE\ U\ p]$

$AX p \equiv \neg EX(\neg p)$

$AG p \equiv \neg EF(\neg p) \equiv \neg E(TRUE\ U(\neg p))$

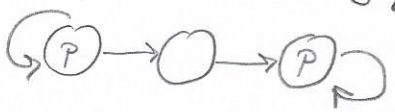
$AF p \equiv A(TRUE\ U\ p) \equiv \neg EG(\neg p)$

$A(p U q) \equiv E[(\neg q) U \neg(p \vee q)] \vee EG(\neg q)$

(9.) LTL FGp i CTL AF(AG p) usporedba

FGp - konačno globalno (stalno) p

AF(AG)p - na svim putevima uvijek dolazimo konačno do stanja iz kojega je dalje stalno p = T



10. a) Uvijek vrijedi: ako je crveno svjetlo onda svjetli dok se ne upali žuto

$AG[crveno \Rightarrow A(crveno \, U \, žuto)]$

b) Uvijek vrijedi: ako je crveno svjetlo tada u slijedeća dva stanja ne smije biti zeleno

$AG[crveno \Rightarrow AX(\neg zeleno \wedge AX(\neg zeleno))]$

11. a) Prvi i četvrti filozof nikad ne mogu jesti istovremeno

$AG \neg(E1 \wedge E4)$

b) Drugi filozof uvijek jede prvi

$\neg(E1 \wedge E3 \wedge E4 \wedge E5) \, AU \, E2$ – ne vrijede dok se ne ostvari E2. Mislim da je moglo A ispred svega

12

a) $AF \, R$ ✓

-za svaki put konačno ćemo doći do stanja u kojem vrijedi R

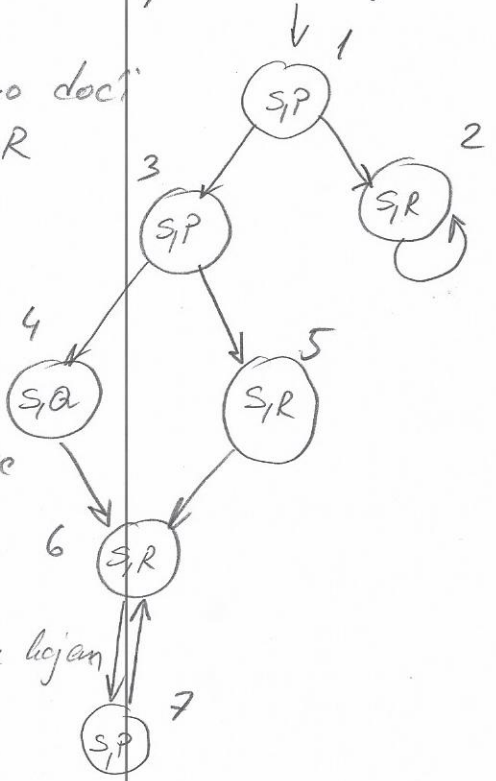
b) $AGEF \, p$ ✗

-za svaki put za sva stanja postoji put u kojem ćemo konačno doći do p → ne zbog stanja 2 gdje zapnemo i ne vrijedi p

c) $AGEF \, R$ ✓

-uvijek možemo doći do stanja u kojem vrijedi R

d) $E[PUE(RUP)]$ ✓



(13.) $EG\ r$

$z_0 = \{0,1,2,3,4\}$, $Q(r) = \{1,3,4\}$

$z_{k+1} = Q(r) \cap R^{-1}(z_k)$

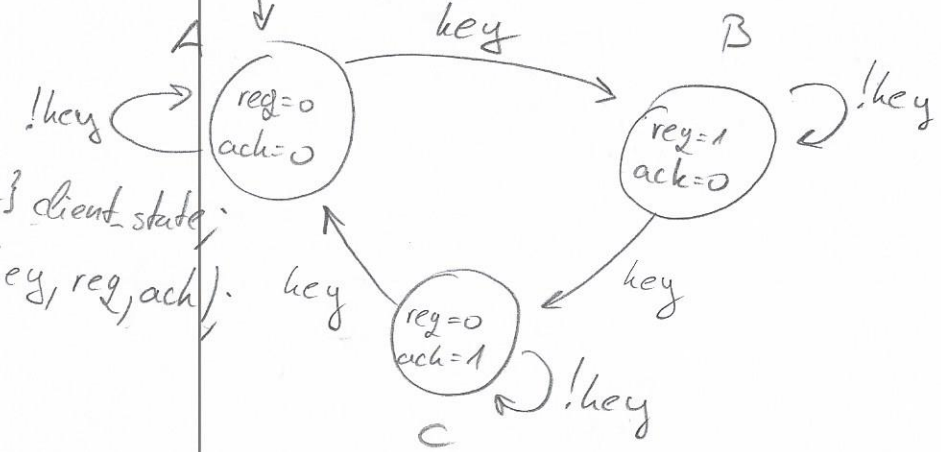
$R^{-1}(z_0) = \{0,1,2,3,4\}$

$z_1 = \{1,3,4\} \cap \{0,1,2,3,4\} = \{1,3,4\}$

$R^{-1}(z_1) = \{0,1,2,3,4\}$

$z_2 = \{1,3,4\} \cap \{0,1,2,3,4\} = \{1,3,4\}$

(14.)




```
typedef enum {A,B,C} client_state;
module client (clk, hey, req, ack);
input hey;
input clk;
output req;
output ack;
client_state reg state;
wire hey, req, ack;
assign req = [state = A && hey];
assign ack = [stat = B && hey];
initial state = A;
always @(posedge clk) begin
        case (state)
            A: if (hey) state = B;
            B: if (hey) state = C;
            C: if (hey) state = A;
        end case
end
end modale
```

(15.) Ima primjera na predavanju + prvi labos izgleda.

15. ima prijavu na predavanjima + 1. Cabo

1.MI 2009/2010

(1.) NABROJATI FORMALNE METODE, U KOJU SPADA ASM?

FORMALNA SPECIFIKACIJA (ASM)
FORMALNA SINTEZA (SDL)
FORMALNA VERIFIKACIJA (SAT)

(2.) NABROJATI TIPOVE SHEMA KOD Z MODELA

1) STATE SHEMA - globalne izjave o sustavu
2) OPERATION SHEMA - opisuje učinak određenih operacija koje mijenjaju stanje podataka u sustavu
3) OBSERVATION SHEMA - opisuje dohvat informacija, podaci u sustavu se ne mijenjaju

(3.) LAGANE, SREDNJE I TEŠKE TEHNIKE, KOJE SU ISPRAVNE, A KOJE KOMPLETNE

VRLO LAGANE - Neispravne i nekompletne
SREDNJE TEŠKE - Ispravne i nekompletne
TEŠKE - Ispravne i kompletne

(4.) STO JE LOGIČKA POSLJEDICA
Skup formula Γ implicira ili povlači formulu ω ako je svaki model od Γ ujedno model od ω. Formula ω je tada logička posljedica skupa formula Γ

(5.) KAKVA SE PROPOZICIJSKA LOGIKA S OBZIROM NA ISPRAVNOST I KOMPLETNOST

Propozicijska logika je ispravna, kompletna i odrediva jer operira s konačnim skupom simbola

(6.) ŠTO JE SAT PROBLEM?

Tražimo model skupa formula Γ (interpretaciju koja evaluira sve formule u skupu Γ istinito)

(7.) PREDIKATNA LOGIKA – NE POSTOJI CIGLA KOJA JE NA CIGLI KOJA JE TAKOĐER NA CIGLI

$C(x)$ – x je cigla

$NA(x,y)$ – x stoji na y

$$\neg\exists x \exists y \exists z\,(C(x) \wedge C(y) \wedge C(z) \Rightarrow (NA(x,y) \wedge NA(y,z)))$$

(8.) PREDIKATNA LOGIKA – AKO JE CIGLA NA ~~CIGLI~~ TADA ~~NIJE~~ NA STOLU

$C(x)$ – x je cigla

$STOL(x)$ – x je na stolu

$NA(x,y)$ – x je na y

$$\forall x \forall y \forall z\,[(C(x) \wedge C(y) \wedge NA(x,y)) \Rightarrow \neg STOL(x)]$$

(9.) SVATKO VOLI NEKOGA I NITKO NE VOLI SVAKOGA

$VOLI(x,y)$ – x voli y

$$\forall x \exists y [VOLI(x,y)] \wedge \neg\exists x \forall y [VOLI(x,y)]$$

(10.) NAPISATI FORMULU KORISTEĆI JEDNAKOST $=(x,y)$

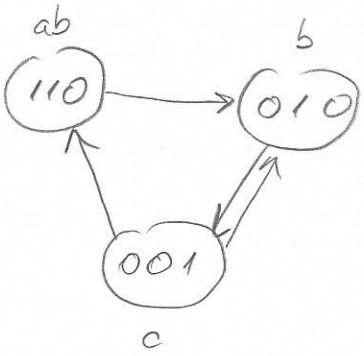
PERO IMA BAREM DVIJE SESTRE

$SESTRA(x, PERO)$ – x je Perina sestra

$$\exists x \exists y (SESTRA(x, PERO) \wedge SESTRA(y, PERO) \wedge \neg =(x,y))$$

(13)

	a	b	c	a	b	c	
	0	0	0	0	0	1	
	0	0	1	0	1	0	1 1 1 0
	0	1	0	0	0	1	
	0	1	1	0	1	0	1 1 1 0
	1	0	0	1	0	1	
	1	0	1	1	1	0	1 0 1 0
start →	1	1	0	0	1	0	
	1	1	1	0	1	1	



$AG(b \rightarrow EX(b \mid c))$ ✓

Na svim putevima, u svim stanjima ako je b tada postoji put za koji je u sljedećem stanju b|c

$AG(c \mid b)$ ✓

Na svim putevima u svim stanjima je c ili b

$AG(AF(a \wedge b))$ NE

- ne jer možemo zapeti između stanja c i b

$EX(AX(c))$ ✓

- postoji put (iz inicijalnog stanja) za koji vrijedi da iz sljedećeg stanja za sva sljedeća stanja vrijedi c

$EF(EG(b \mid c))$ ✓

- postoji put kojim ćemo konačno doći u stanje iz kojeg postoji put za koji u svim stanjima vrijedi b ili c

$EX(c)$ NE

- ne jer iz inicijalnog stanja nemamo sljedeće stanje u kojem vrijedi c