

(*) Prädikation \leftrightarrow CNF

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

~~Prädikation~~

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

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

$$= \underbrace{(P \vee \neg P \vee \neg R \vee Q)}_1 \wedge$$

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

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

$$= \boxed{(Q \vee \neg P \vee \neg R) \wedge (\neg R \vee \neg P \vee Q)}$$

$$\boxed{\overline{Q \vee \neg P \vee \neg R}}$$

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

$$= \frac{((P \wedge \neg P) \vee (P \wedge Q)) \wedge ((\neg Q \wedge \neg S) \vee (\neg S \wedge S))}{0}$$

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

$$= P \wedge Q \wedge \neg Q \wedge \neg S$$

$$= \text{False} \Leftrightarrow S \text{ ist wortlos.}$$



Sveti prof. je zaposleni bočna jednog fakulteta, a predaje na jednom drugom

$\text{PROF}(x) = x \text{ je prof.}$

$F(x) = x \text{ je fakultet}$

$Z(x, y) = x \text{ je zaposlen na } y$

$\text{PREDATE}(x, y) = x \text{ predaje na } y$

$= (x, y) = x \text{ je pedavus } y$

$$\forall x \left[\text{PROF}(x) \Rightarrow \exists y \left[F(y) \wedge Z(x, y) \wedge \exists z (F(z) \wedge Z(x, z) \wedge T = (y, z)) \right] \wedge \exists y (F(y) \wedge P(x, y)) \right]$$

\Rightarrow Unikum programu ~~definisan~~ definisan su najmanje dva razlicita rastreka

$P(x) = x \text{ je program}$

$R(x) = x \text{ je rastrek}$

$\text{DEF}(x, y) = x \text{ je definisan u } y$

$= (x, y) = x \text{ je pedavus } y$

$$\exists x \exists y \exists z (P(x) \wedge R(y) \wedge R(z) \wedge D(y, x) \wedge D(z, x) \wedge T = (y, z))$$

\Rightarrow Sveti se voli svakog, i niti se voli svakog

$\text{VOLI}(x, y) = x \text{ voli } y$

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

4.) a) DA

b) NE (zbyt α)

c) DA

d) NE (CTL^*)

e) NE (nich jedna)

f) DA

5.) a) AG A($\neg p \vee g$)

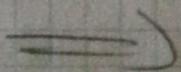
b) EF(p \wedge AX($\neg p \wedge \underbrace{AX(\neg p)}_{\text{nemjedn } p}$) \vee $\underbrace{\text{onou daje me}}_{\text{mijedi } p}$

c) $p \wedge EX \neg g$

6.) a) $G !(p \wedge \neg g) \equiv \neg F(p \wedge \neg g)$

b) $F(p \wedge GFp)$

c) $G(p \Rightarrow (g \vee \neg p))$



7)

$$Z_0 = \emptyset$$

$$Z_{l+1} = Q(r) \cup (Q$$

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

slup skup
gde je vnijski

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

$$= \{1, 3, 4\} \neq Z_0$$

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

$$= \{0, 1, 3, 4\} + Z_1$$

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

$$= \{0, 1, 3, 4\} = Z_2 \Rightarrow Z_3 \text{ je fix point}$$

(?)

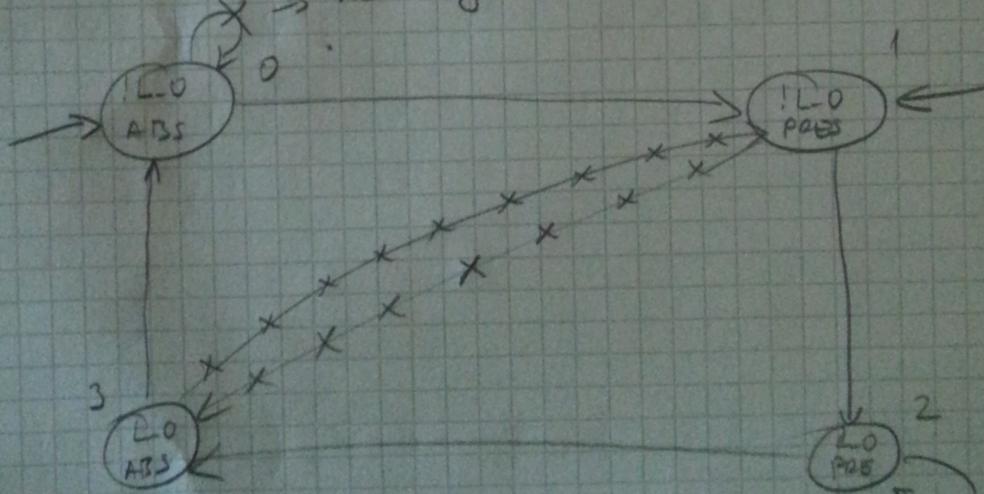
line-on
absent

line-on
absent

line-on
present

!line-on
present

ne abog JUSTICE



a) DA, 2. stanje

b) DA (\cup 2 \rightarrow 3 \wedge 4 \wedge)

⑩ `typedef enum { START, LISTEN, BUSY } conn-state;`

module server-connection (clk, listen, established, stop,
ready, reset);

Healthy

 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
 case (state)

 START : if (listen) state = LISTEN

 else if (reset) state = START (reset
 now)

 LISTEN : if (reset) state = START

 else if (established) state = BUSY

 else if (listen) state = LISTEN (reset now)

 BUSY : if (stop) state = LISTEN

 else if (reset) state = START

 else if (established) state = BUSY (reset now)

 end case

end module
end end

\Rightarrow MUX RG \rightarrow output 1^e v

strange
output

no ppt

\rightarrow da je ready wire, oder bij v always
blokk nation state = LISTEN (apr.) small
begin . prompter ~~v~~ ready and

⑪

~~read - blif - mr~~ server - connection. mr
print - models
init - verify
~~read - fairness~~ server - connection. fair
module - check - i server - connection. ct