

## ① DEFINIRANJE TEOREMA DEDUKCIJE

- a) Formula  $\phi$  je logička posljedica formule  $\psi$  tj.  $\psi \models \phi$ , ako i samo ako je formula  $(\psi \Rightarrow \phi)$  tautologija (uvijek istinita)
- b)

## ② Dio Verilog koda glasi:

```
always @(posedge clk) begin
  a=5;
  b=2;
  bc=a;
  v=b;
end
```

Koja je vrijednost pridružena varijabli v neposredno po izvođenju zadnje naredbe

V=2

## ③ Izrazi CTL formule pomoću operatora AU

a)  $AF p \equiv A(TRUE \cup p)$

inače...

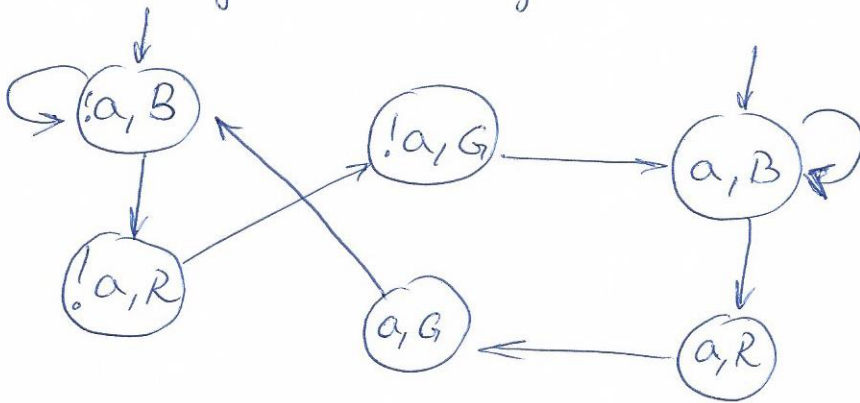
$$AF p \equiv \neg EG(\neg p)$$

b)  $EG p \equiv \neg A(FALSE \cup \neg p)$

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

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

- ⑤ Za zadanu Kripke strukturu potrebno je napisati odgovarajući NuSMV kod i odrediti istinitost CTL specifikacija promatrajući iz dva početna stanja (mora vrijediti za oba)



- a) CTLSPEC  $AG AF (!a \ \& \ st = G)$   
 b) CTLSPEC  $(a \ \& \ EG (st = R))$

VAR

a: boolean

st: {R, G, B}

ASSIGN

init(a) := {FALSE, TRUE}

init(st) := B

next(a) := case

(a = FALSE & st = B): FALSE;

(a = FALSE & st = R): FALSE;

(a = FALSE & st = G): TRUE;

(a = TRUE & st = B): TRUE;

(a = TRUE & st = R): TRUE;

(a = TRUE & st = G): FALSE;

esac;

next(st) := case

(a = FALSE & st = B): {B, R};

(a = FALSE & st = R): G;

(a = FALSE & st = G): B;

(a = TRUE & st = B): {B, R};

(a = TRUE & st = R): G;

(a = TRUE & st = G): B;

a) vrijedi

b) ne vrijedi

6. Detaljno opišite funkcioniranje svake od navedenih Promela naredbi:

a) mtype:

mtype = {RDY, DATA, ACK, REQ}

mtype deklarira tip poruke, ovdje su to četiri tipa  
rdy, data, ack i req

b) chan

chan b2a = [2] of {bit, byte, bit, bit}

Deklarira kanal pod imenom b2a kapaciteta kanala 2 (da je bilo  
0, označi sinkronu izmjenu poruka), čija je struktura poruke {bit, byte, bit, bit}

c) timeout:

```
do
  :: ch1? ACW;
  :: ch2? BWL;
  :: timeout → goto end_TRAP;
od;
```

do - nedeterministička petlja  
if - nedeterministička selekcija  
simbol! - predaja  
simbol? - prijem

prijem/predaja označe stvaranje/  
užimanje poruke u kanal koji opisuje  
komunikaciju između dva procesa

7. Napisati pseudokod za ite(f, g, h) s unique i computed tablicama

Terminal cases:  $(0, g, f) = (f, f, g) = f$   
 $ite(f, g, g) = g$

ite(f, g, h)

```
if (terminal case) {
  return result;
} else if (computed-table has entry (f, g, h)) {
  return result;
} else {
  let v be the top variable of (f, g, h);
```

```
  f̃ ← ite(fv, gv, hv);
  g̃ ← ite(fv, gv, hv);
  if (f̃ equals g̃) return g̃;
  R ← find-or-add-unique-table(v, f̃, g̃);
  insert-computed-table({(f, g, h)}, R);
  return R;
}
```

⑧ Kolika je složenost za bdd  $ite(f, g, h)$

Bez uporabe izračunske tablice

- jedan pristup jedinstvenoj tablici (bez rekureije) = konstantno vrijeme
- rekursivni poziv (ako nije završni) traži 2 nova pristupa tablici
- vrijeme izvođenja je eksponencijalno prema broju varijabli

Uz uporabu izračunske tablice (neograničena memorija)

- neka je  $|f|$  broj čvorova u ROBDD-u za  $f$
- za jedinstvenu kombinaciju  $(f, g, h)$  funkcija  $ite(f, g, h)$  se poziva jednom
- općenito  $ite(f, g, h)$  se poziva  $|f| \cdot |g| \cdot |h|$  puta te je složenost  $O(|f| \cdot |g| \cdot |h|)$
- za  $ite$  funkciju s dva operanda (npr. AND, OR, XOR...) složenost je  $O(|f| \cdot |g|)$  u najgorem slučaju

⑨ Napisati pseudokod za BDD  $H$  uz zadan BDD  $R$

BDD  $H$  (BDD  $A$ , BDD  $R$ ) {

  return  $bdd\_exist(\underline{s}, bdd\_and(A, R))$  }

$\underline{s}$  - vektor svih stanja u zadanom početnom skupu

$bdd\_exist()$  - funkcija isključuje iz rezultata početna stanja  $\{s\}$

$bdd\_exist(s, f)$ :  $\exists_s f = f_s + f_{s'} = ite(f_s, 1, f_{s'})$



(10) Uporaba Shannonovog teorema do varijable  $a$ . Napiši oba faktora  
a zatim maksimalno pojednostavi rezultat

$$F(a,b,c) = a'bc + ab'c + abc' + abc$$

Shannon  $f = xf_x + x'f_{x'}$

$$f_a = b'c + bc' + bc = c(\underbrace{b' + b}_1) + bc' = c + bc' = b + c$$

$$f_{a'} = bc$$

$$f = a(b+c) + a'(bc) = ab + ac + a'bc$$

(11) Napiši AND, OR, XOR, NAND u ite obliku

AND  $fg$   $ite(f, g, 0)$

OR  $f+g$   $ite(f, 1, g)$

XOR  $f \oplus g$   $ite(f, \bar{g}, g)$

NAND  $\overline{fg}$   $ite(f, \bar{g}, 1)$

NOR  $\overline{f+g}$   $ite(f, 0, \bar{g})$

EQUIV  $f \Leftrightarrow g$   $ite(f, g, \bar{g})$

IMPLIK.  $f \Rightarrow g$   $ite(f, g, 1)$

NOT  $\bar{f}$   $ite(f, 0, 1)$

⑫  $F(x_1, x_2, x_3) = (x_1 + x_2)(x_1 + x_3)(x_2 + x_3)(x_1' + x_2')$

a)

$x_1$	$x_2$	$x_3$	$x_1 + x_2$	$x_1 + x_3$	$x_2 + x_3$	$x_1' + x_2'$	$f$
0	0	0	0	0	0	1	0
0	0	1	0	1	1	1	0
0	1	0	1	0	1	1	0
0	1	1	1	1	0	1	0
1	0	0	1	1	0	1	0
1	0	1	1	0	1	1	0
1	1	0	1	1	1	0	0
1	1	1	1	0	0	0	0

$x_1 x_2$	00	01	11	10
$x_3$				
0				
1		1		1

$f = x_1' x_2 x_3 + x_1 x_2' x_3$

a)  $x_1 < x_2 < x_3$

b)  $x_3 < x_1 < x_2$

c)  $x_3 < x_1 < x_2$  s komplementima

$f_{x_1} = x_2' x_3$

$f_{x_1'} = x_2 x_3$

$f_{x_1 x_2} = 0$

$f_{x_1 x_2'} = x_3$

$f_{x_1' x_2} = x_3$

$f_{x_1' x_2'} = 0$

$f_{x_3} = x_1' x_2$

$f_{x_3'} = 0$

$f_{x_3 x_1} = 0$

$f_{x_3 x_1'} = x_2$

