# Blic 13/14

1. Kripke struktura zadana je skupom stanja S, binarnom relacijom R i funkcijom označavanja L. Neka je:

∀s ∈S (∃t ∈S | (s,t) ∈R)

Što je ispravno?

1. Skup {s} je slika skupa {t} pod relacijom R.
2. **Skup {s} je pred-slika skupa {t} pod relacijom R.**
3. Skup {t} je slika skupa {s} pod inverznom relacijom

2. Za kripkie strukturu vrijedi:

1. R-1 (S) = ∅, R-1 (∅) = ∅
2. R-1 (S) = ∅, R-1 (∅) = S
3. **R-1 (S) = S, R-1 (∅) = ∅**
4. R-1 (S) = S, R-1 (∅) = S

3. Odredi ispravan izraz za Kripke strukturu:

1. EG fi = fi V EX EG fi
2. **AF fi = fi V AX AF fi**
3. E[fi U gam] = gam & (fi & EX E(fi U gam))

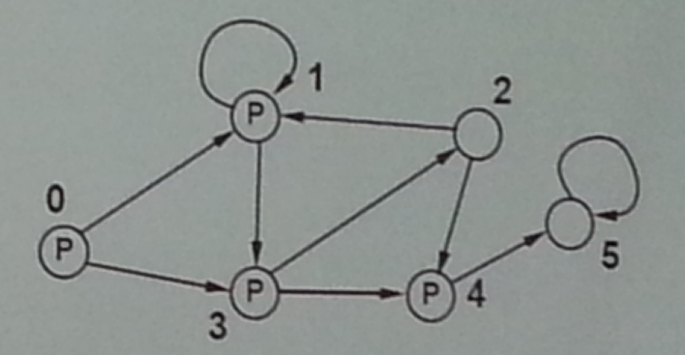
4. Neka postoji monotona funkcija koja preslikava sve podskupove

F: P(S) → P(S)

Ispravno je:

1. **Primjenjujući F do n+1 puta na cijeli skup S slijedi najmanja čvrsta točka.**
2. Primjenjujući F do n+1 puta na prazan skup ∅ slijedi najveća čvrsta točka.
3. Primjenjujući F do n+1 puta na prazan skup ∅ slijedi najmanja čvrsta točka.

5. Zadana je kripkie struktura



Što je ispravno?

1. **R-1 {1, 2} ={0, 1, 2, 3}**
2. R-1 {1, 2} ={0, 3}
3. R-1 {1, 2} ={0, 1, 3}

6. Za logičku funkciju:

F = ac + a'c'

Kofaktor po varijabli a' (t.j. po komplementu a) je:

1. fa' = ac
2. fa' = ac'
3. fa' = c
4. **fa' = c'**
5. fa' = a'c'

7. Logičku disjunkciju (f V g) ite operator implementira kao:

1. ite(f, 1, g)
2. ite(f, 1, 0)
3. ite(f, 0, g)
4. ite(f, g, 1)

8. Pri implementaciji procedure oblikovanja ROBDD dijagrama upotrebljava se jedinstvena tablica (engl. unique table) s kolizijskim lancem. Tu se upisuje konačan rezultat izračunavanja neke ite operacije. t.j. ite(f, g, h). Parametri za upis u tablicu su: f , g, h.

1. Ispravno
2. **Neispravno**

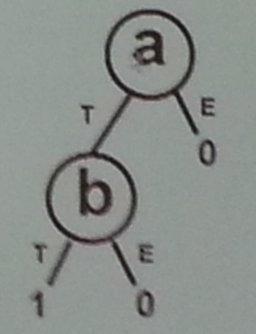
9. Što je ekvivalentno ite operatoru ite(f, g, 0)

1. Ite(f, 0, g)
2. **Ite(g, f, 0)**
3. Ite(g, 0, f)
4. Ite(0, f, g)

10. Ispravan izraz za egzistencijsku ili univertalnu apstrakciju je:

1. ∃s f = fs x fs'
2. **∃s f = fs + fs'**
3. ∀s f = fs + fs'

11. ROBDD na slici predstavlja



1. a V b
2. a V b'
3. **a & b**
4. a & b'

12. Ako postoji ROBDD za neku logičku funkciju, provjera da li je zadovoljiva (engl. Satisfiable) može se odrediti:

1. u eksponencialnom vremenu prema broju čvorova
2. u polinomnom vremenu prema broju čvorova
3. u linearnom vremenu prema broju čvorova
4. **u konstantnom vremenu**

# Blic 14/15

1. Skupove s konačnim brojem elemenata i binarne relacije nad tim skupovima moguće je prikazati Booleovim funkcijama i to najprije:

1. izgradnjom BDD-ova nad skupovima
2. **kodiranjem elemenata skupova odnosno relacija**
3. izdvajanjem minterma ili maksterma

2. Zapis Booleove funkcije f= x1x2' + x1x2 dan je u obliku:

1. Tablice istinitosti
2. **Sume minterma**
3. Produkata maksterma
4. **Sume produkata**

3. Koje od sljedećih izjava **ne** vrijede za binarne dijagrame odlučivanja (ROBBD-ove)?

1. Ostvaruju kanonski prikaz Booleovih funkcija
2. **Za bilo koju Booleovu funkciju , broj čvorova u BDD-u je polinoman u ovisnosti o broju varijabli**
3. Omogućuju efikasnu provjeru ekvivalentnosti Booleovih funkcija
4. Provjera tautologije je u konstantnom vremenu

4. Negativni kofaktor funkcije f = x y'z + x y + y' po varijabli cijepanja y jednak je:

1. **1**
2. 0
3. Xz
4. x

5. Koju logičku funkciju ostvaruje ITE-operator ite(f, 1, g)?

1. AND
2. **OR**
3. XOR
4. 1

6. Pri implementaciji ROBDD-a ITE algoritmom koriste se dvije tablice: jedinstvena i izračunska. Što se pohranjuje u jedinstvenoj tablici?

1. **Struktura koja se sastoji od indeksa vršne varijable, indeksa THEN i ELSE strane i pokazivača na sljedeći element kolizijskog lanca**
2. Struktura koja se sastoji od indeksa vršne varijable, indeksa THEN i ELSE strane (bez kolizijskog lanca)
3. Svaki izračunati BDD, uključujući i kopije ranije izračunatih
4. Jedinstvena tablica sadrži samo popis završnih (trivijalnih) ROBDD-ova

7. Kanoničnost ROBDD-a pri uvodenju komplementiranih lukova ostvarena je konsenzusom prema kojem:

1. **THEN luk ne smije imati komplement**
2. ELSE luk ne smije imati komplement
3. je ELSE luk uvijek komplementiran
4. Kanoničnost u slučaju uvodenja komplementiranih lukova nije ostvarena.

8. Pri simboličkoj provjeri modela korištenjem ROBDD-ova koristi se teorija čvrste točke (engl. fix-point). Koju formulu vremenske logike ostvaruje iterativna relacija nad ROBDD-ovima

**Zk+i : = bdd\_or (q, bdd\_and (p , H-1(Zk) ) ;**

1. EX p
2. EF q
3. **E (p U q)**
4. A(p U q)

9. Osnovni algoritam **DPLL** za SAT-problem koristi dvije heuristike u svakom koraku izvođenja. To su:

1. Učenje novih klauzula kroz konflikt
2. **Propagacija jedinične klauzule**
3. Ustanovljavanje jedinstvene implikacijske točke
4. **Uklanjanje klauzula sa čistim literalima**

10. Učenje novih klauzula kroz konflikt i nekronološki povrat na prethodnu varijablu najbitniji su noviteti SAT-rješavača:

1. DPLL
2. MiniSAT
3. **GRASP**
4. Chaff

11. Ako je varijabla y forsirana (implicirana) u klauzuli K = ( ¬x V y V ¬z) tijekom propagacije Booleovih ograničenja, tada je pridruživanje prethodnika varijable y (A(y)) jednako:

1. A(y) = {x = 0, z = 0}
2. A(y) = {x = 1, z = 0}
3. A(y) = {x = 0, z = 1}
4. **A(y) = {x = 1, z = 1}**

12. SAT-rješivač Chaff koristi vrlo učinkovitu propagaciju Boolevih ograničenja kod koje se uvode promatrani literali (engl. watched literals). Koliko se literala po klauzuli od p literala promatra u svakom trenutku?

1. p -1
2. 3
3. **2**
4. 1

# Blic 15/16

1. nema

2. Što od navedenog nije način definiranja kašnjenja u programskom jeziku Verilog?

1. Wait (f==0) q=3;
2. Nešto čudno, kratko
3. Always @posedge (var) w=4;
4. **After (d) a=1;**

3. Skupove s konačnim brojem elemenata i binarne relacije nad tim skupovima moguće je prikazati Booleovim funkcijama i to najprije :

1. **Kodiranjem elemenata skupova odnosno relacija**
2. Izgradnjom BDD-ova nad skupovima
3. Računanjem produkta suma ili sume produkata

4. ROBDD ima svojstvo da u njemu ne postoji čvor s istim čvorom djetetom za oba grananja (then i else). Koje od navedenih tvrdnji vrijede?

1. **ROBDD je manji od BDD koji takve čvorove ima**
2. Takvi čvorovi se zovu izomorfni grafovi
3. Uklanjanjem takvih čvorova uvijek ostaje samo veza prethodnog čvora na završni čvor 0
4. **Kanoničnost prikaza Booleove funkcije je zadržana kod ROBDD-a**

5. Negativni kofaktor funkcije f=xy'z + xy + y' po varijabli cijepanja x jednak je:

1. 1
2. 0
3. y'z
4. **y'**

6. Koju logičku funkciju ostvaruje ITE-operator **ite**(f, g, g')?

1. XOR
2. **XNOR (ekvivalencija)**
3. AND
4. OR

7. Koje sve tvrdnje vrijede za ITE-algoritam?

1. **Gradi ROBDD**
2. **Koristi jedinstvenu i izračunsku tablicu**
3. Složenost mu je uvijek eksponencijalna prema broju varijabli
4. **Rekurzivan je**

8. Izračun skupovima logičke formule EG p, zadan kao Q(EG p) = Q(p) ∩ Q(EX EG p) u BDD varijanti glasi:

1. Zk+i : = bdd\_or (p, bdd\_and (p , H-1(Zk))) ;
2. Zk+i : = bdd\_and (p, bdd\_or (p , H-1(Zk))) ;
3. **Zk+i : = bdd\_and (p, H-1(Zk)) ;**
4. Zk+i : = bdd\_or (p , H-1(Zk)) ;

9. Osnovi algoritam **DPLL** za SAT-problem koristi dvije heuristike u svakom koraku izvođenja. To su:

1. Učenje novih klauzula kroz konflikt
2. **Uklanjanje klauzula s čistim literalima**
3. **Propagacija jedinične klauzule**
4. Ustanovljavanje jedinstvene implikacijske točke

10. Kod SAT-rješavača GRASP, broj 1 iza oznake @ u primjeru x = 0@2 označava:

1. Broj vrijednosti koje varijabla x može poprimiti
2. **Razinu odluke na kojoj je varijabli x dodijeljena vrijednost**
3. Redni broj varijable x u grafu implikacija
4. Redni broj klauzule u kojoj se pojavljuje varijabla x

11. Najveće poboljšanje algoritma Chaff u odnosu na algoritam GRASP ostvareno je učinkovitijim postupkom propagacije Booleovih ograničenja u klauzulama:

1. **DA**
2. NE

12. Rezolucijom klauzula (a V ¬b V c) i (¬a V c V d) po varijabli a dobiva se klauzula:

1. 0 (kontradikcija)
2. 1 (tautologija)
3. C
4. **¬b V c V d**