

# FMUOS - 1. KPZ

April 12, 2021

# 1 Pitanja

Pitanje:

Ako i samo ako postoji algoritam koji će u konačnom vremenu odrediti ili ne dedukciju omega (dati u konačnom vremenu da je omega dedukcija ili da omega nije dedukcija), tada je formalan sustav:

Odgovor:

**odrediv (engl. Decidable)**

Pitanje:

Kvantifikator "za svaki" ispravno se koristi s (1), a kvantifikator "postoji" ispravno se koristi s (2).

Odgovor:

**(1) implikacijom; (2) konjunkcijom**

Pitanje:

Java Pathfinder koristi djelomično smanjenje poretka (engl. partial order reduction) kako bi riješio (smanjio) problem:

Odgovor:

**Eksplozije broja stanja**

Pitanje:

Izbacite uljeza:

Provjera ekvivalentnosti

**Inspekcija koda**

Dokazivanje teorema

Provjera modela

Pitanje:  
Na prikazanom dijagramu, za početno stanje S, dan je primjer vremenske formule:

[illegible]

Slika 1: Dijagram

Odgovor:

$$AX_p$$

EG p

$$\text{AG } \overline{\text{EF}} \text{ }_p$$

AF p

Pitanje:

Preslikajte rečenicu prirodnog jezika: "Ako je in istinit u početnom trenutku, out će biti istinit u nekom budućem trenutku" u formalizam logike LTL

Odgovor:

$X(\text{in} \Rightarrow G \text{ out})$

$G(\text{in} \Rightarrow F \text{ out})$

**$\text{in} \Rightarrow F \text{ out}$**

$G(\text{in} \Rightarrow XG \text{ out})$

Pitanje:

Java PathFinder može se kretati unaprijed ili unazad kroz naš program pri izvođenju jer:

Odgovor:

**a. PathFinder koristi strategije pretraživanja stanja**

b. Svaki program za otklanjanje pogrešaka (engl. debugger) to može

**c. PathFinder se izvodi kao virtualni stroj iznad Javinog virtualnog stroja kako bi istražio naš program**

d. Slušači omogućuju simulaciju kretanja kroz stanja

Pitanje:

Što od navedenog **nije** razina sinteze sklopovlja:

Odgovor:

Postavljanje i usmjeravanje

Sinteza visoke razine

**Sistemska analiza**

**Sistemska sinteza**

Logička sinteza

Pitanje:

U Verilogu, koje se od navedenih naredbi uzimaju u obzir pri sintezi:

Odgovor:

- a. #n
- b. @posedge(var) q = 1;**
- c. wait(var == 0) q = 1;
- d. sleep

Pitanje:

U Verilogu, ključnu riječ **assign** koristimo pri pridruživanju vrijednosti izlaznim varijablama koje su po svojoj vrsti:

Odgovor:

- a. rand
- b. opt
- c. reg
- d. wire**

Pitanje:

Pomoću izraza na slici, za sve podskupove X i Y u S provjerava se:

$$(X \subseteq Y) \implies (F(X) \subseteq F(Y))$$

Odgovor:

- a. Fiksna točka
- b. Tranzitivnost relacija
- c. Monotonost funkcija**
- d. Teorem Knaster-Tarski

Pitanje:

Što od navedenog **nije** svojstvo konfiguracijskog objekta u sustavu Java PathFinder?

Odgovor:

- a. Prenosi se odozgo prema dolje u hijerarhijskom procesu
- b. Zasnovan je na nizovima znakova
- c. Pohranjuje stanja dretvi**
- d. Proširiv je

Pitanje:

Što od navedenog **nije** uobičajena konfiguracijska datoteka koju koristi sustav Java PathFinder?

Odgovor:

- a. include.properties**
- b. site.properties
- c. jpf.properties
- d. Verifikacija.jpf

Pitanje:

Značenje ključne riječi DEFINE u NuSMV je:

Odgovor:

**Definicija određene varijable koja se ponaša kao makro instrukcija.**

Pitanje:

SAT-problem je za klauzule koje se sastoje do najviše 3 literala:

Odgovor:

**NP-kompletnan**

Pitanje:

U vremenskoj logici CTL, logički izraz  $AG\ EF\ p$  znači:

Odgovor:

**Iz svakog stanja sustava u koje se dođe iz početnog stanja moguće je barem na jednom putu konačno doći do stanja u kojem vrijedi  $p$**

Pitanje:

U Verilogu, vrsta varijable koja se koristi kada je potrebno spremati vrijednosti u vidu memorijskog elementa, naziva se:

Odgovor:

**reg**

Pitanje:

Linearna vremenska struktura LTL-logike dana je trojkom:

Odgovor:

$\pi = (S, x, L)$

Pitanje:

U NuSMV-u, stil ograničenja (engl. constraint) koristi naredbe INIT, INVAR i:

Odgovor:

**TRANS**

Pitanje:

Preslikajte rečenicu prirodnog jezika: "Kadgod je in istinit, out će biti istinit u nekom budućem trenutku" u formalizam logike LTL.

Odgovor:

- a.  $G(in \Rightarrow XG out)$
- b.  $G(in \Rightarrow F out)$**
- c.  $X(in \Rightarrow G out)$
- d.  $in \Rightarrow F out$

Pitanje:

U NuSMV-u, ako želimo pridružiti trenutačnu vrijednost (u svakom trenutku) varijabli turn iz skupa {ready,busy} to ćemo ispravno napraviti pomoću izraza:

Odgovor:

- a. `turn := {ready,busy};`**
- b. `turn = {ready,busy};`
- c. `next(turn) := case TRUE:{ready,busy}; esac;`
- d. `next(turn) := {ready,busy};`

Pitanje:

Kod optimizacije oblikovanja na razini logičkih vrata (netliste), potrebno je nakon svakog koraka optimizacije napraviti:

Odgovor:

**Provjeru ekvivalentnosti**

Pitanje:

Coq, Isabelle/HOL i Verifast primjeri su alata za koju metodu formalne verifikacije?

Odgovor:

**Dokazivanje teorema**



Pitanje:

Formula  $E(XF p)$  sintaksno je ispravna u logici:

Odgovor:

- a. Formula nije ispravna ni u jednoj navedenoj logici
- b. CTL
- c. CTL\***
- d. LTL

Pitanje:

Formalan sustav je dvojka  $\{\text{gama}, L\}$ , pri čemu je  $L$ :

Odgovor:

- a. Skup pravila zaključivanja**
- b. Interpretacija
- c. Skup modela
- d. Logička posljedica

Pitanje:

Što od navedenog spada u metode formalne verifikacije:

Odgovori:

- a. Ispitivanje kritičnih sustava
- b. Inspekcija koda
- c. UML
- d. Provjera ekvivalentnosti**
- e. Provjera modela**
- f. Dokazivanje teorema**
- g. Provjera tvrdnje**

Pitanje:

Izrazom prikazanom na slici koristimo se kod eksplicitnog izračunavanja skupa stanja preko najveće fiksne točke za formulu vremenske logike:

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

Odgovor:

- a. EF p
- b. EG p**
- c. E(p U q)
- d. EX p

Pitanje:

Pravedan odabir procesa za izvođenje u sustavu NuSMV može se postići navođenjem izraza:

Odgovor:

- a. CTLSPEC AG AF running
- b. FAIRNESS process
- c. JUSTICE !(st = c)
- d. JUSTICE running**

Pitanje:

Z i ASM su primjeri kakvih formalnih metoda (prema Kroeningovoj klasifikaciji):

Odgovor:

- a. Metoda formalne specifikacije**
- b. Metoda formalne sinteze
- c. Metoda formalne verifikacije
- d. Metoda formalne eliminacije

Pitanje:

U vremenskoj logici CTL, logički izraz  $EF\ AG\ p$  znači:

Odgovor:

- a. Iz početnog stanja sustava na svim putovima cijelo vrijeme vrijedi  $p$
- b. Iz početnog stanja sustava postoji put na kojem se konačno dolazi do stanja u kojem dalje stalno vrijedi  $p$**
- c. Iz početnog stanja sustava postoji put na kojem cijelo vrijeme vrijedi  $p$
- d. Iz svakog stanja sustava na svim putovima koji vode od početnog stanja se konačno dolazi do stanja u kojem vrijedi  $p$

Pitanje:

Što spada u metode formalne verifikacije sklopovlja?

Odgovor:

- a. Provjera ekvivalentnosti**
- b. Provjera modela**
- c. Automatizirano dokazivanje teorema**
- d. Provjera tvrdnje**