



1. Zadan je Promela proces *Acquirer*:

```
1 int i=0;
2 while (true) {
3   if (i < 10) {
4     i++;
5   } else {
6     i = 0;
7   }
8 }
```

Koja je vrijednost varijable *i* na kraju izvođenja procesa. Izvedite *i* se naredba u redu *B* (*ack*)?

- (a) $i = 1$, *ack* se ne izvede i ne dolazi do zastoja
- (b) $i = 0$, *ack* se ne izvede jer dolazi do zastoja
- (c) $i = 0$, *ack* se izvede i ne dolazi do zastoja
- (d) $i = 1$, *ack* se ne izvede jer dolazi do zastoja

2. Koja od navedenih formula nije sintaksno ispravna u logici LTL:

- (a) $p \cup q \wedge r$
- (b) $GF(p \vee Fq)$
- (c) $XXGp$
- (d) $G(p \Rightarrow Fq)$

3. U jeziku Promela naredba $a \{ b$:

- (a) Šalje poruku *b* preko komunikacijskog kanala *a*
- (b) Definiira asinkroni produkt procesa *a* i procesa *b*
- (c) Definiira sinkroni produkt procesa *a* i procesa *b*
- (d) Šalje poruku *a* preko komunikacijskog kanala *b*

4. Zadan je ekspanzirani asinkroni produkt (σ^a) s vrijednostima varijabli:

$x = 2; x = 1; x = 5; x = 4; x = 7; x = 9; x = 10; x = 13;$
 $x = 12; x = 11;$

Odredite za koji od ponuđenih odgovora je LTL formula $\Box (Gp \rightarrow \text{"globally"} p)$ istinita

- (a) $p \equiv (x > 0)$
- (b) $p \equiv (x = 7)$
- (c) $p \equiv (x \geq 10)$
- (d) $p \equiv (x < 7)$

om CTLSPEC AG AF (turn = one) u

vodimo specifikaciju koju treba provjeriti
 vodimo ograničenje koje treba uvijek vrijediti
 varijablu *turn*

niramo makro instrukciju

iramo pravednost koja treba biti ispoštovana

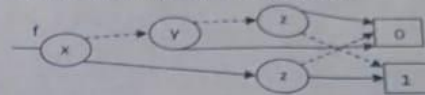
5. Ispravno, netočno ili sklopitično redukcijama se

- (a) $req = !turn, Fack$
- (b) $req = !turn, Fack$
- (c) Kod sklopitičnog redukcijama se naredbe ne
- (d) $req = !turn, Fack$

7. Što od navedenoga nije način kako osigurati projekt Java

- (a) Djelomično smanjenje potetka (engl. *partial order reduction*)
- (b) Delegiranje izvođenja metoda javinom virtualnom stroju
- (c) Simboličko izračunavanje skupova stanja
- (d) Provjera podudaranja stanja (engl. *state matching*)

8. Na slici je prikazan (u vodoravnom obliku):



- (a) ROBDD bez komplementiranja lukova
- (b) Osnovni BDD
- (c) ROBDD s neispravnim komplementiranjem lukova
- (d) ROBDD s ispravnim komplementiranjem lukova

9. Koja od navedenih formula nije sintaksno ispravna logici CTL:

- (a) $EF req \wedge AG ack$
- (b) $A(\neg req \cup \neg ack)$
- (c) $E(\neg req \cup \neg ack \wedge req)$
- (d) $E(\neg req \cup A \neg ack)$

10. Prevedite rečenicu prirodnog jezika "Ako u početnom stanju vrijedi ϕ , onda će ϕ nastaviti vrijediti sve dok ne počne vrijediti ψ ." u formulu vremenske logike LTL

- (a) $\phi \Rightarrow X(\phi \cup \psi)$
- (b) $G(\phi \Rightarrow (\phi \cup \psi))$
- (c) $X(\phi \Rightarrow (\phi \cup \psi))$
- (d) $\phi \Rightarrow F \psi$



00:05

00:18





11. Koja vrijednost poprima varijabla y nakon izvršenja strukture A koja izgleda kao u nastavku naredba?
- ```
if (x > 0) {
 y = 2 * x;
} else if (x < 0) {
 y = -1 * x;
} else {
 y = 0;
}
```
- (a)  $y = 2$   
(b)  $y = 0$   
(c)  $y = 3$   
(d)  $y = 4$
12. Zadan je ekspanziran asinkroni produkt ( $\sigma^m$ ) s vrijednostima varijabli:  
 $x = 2; x = 1; x = 3; x = 4; x = 7; x = 9; x = 10; x = 13;$   
 $x = 12; x = 11;$   
Odredite za koji od ponuđenih odgovora je LTL formula  $\Diamond p$  ( $Fp$  - "eventually"  $p$ ) istinita
- (a)  $p = (x == 7)$   
(b)  $p = (x > 64)$   
(c)  $p = (x \geq 256)$   
(d)  $p = (x < 0)$
13. Što od navedenih tehnika ne koristi B-metoda (Event-B) za formalnu sintezu?
- (a) Teoriju skupova kao notaciju za modeliranje  
(b) Rafiniranje za predstavljanje sustava na različitim razinama apstrakcije  
(c) Matematičke dokaze za provjeru konzistentnosti između različitih razina  
(d) Provjeru modela zasnovanu na SAT-rješavačima
- Program koji odlučuje o zadovoljivosti formule redikativne logike prvoga reda u odnosu na pozadinske vrijednosti kao što su teorija jednakosti, cijelih i realnih brojeva, polja, itd. naziva se:
- (a) SMT-rješavač  
(b) ipf-core  
(c) NuSMV  
(d) SAT-rješavač
- Simboličkom registracijom se odnosi na:
- Simboličkom ograničenju puta
- Simboličkom programskom brojevima
- Simboličkom stanju
14. Koja je od navedenih tehnika najčešće korištena za sintezu (FMT) istinita?
- (a) FMT u principu povećava vrijeme izvršenja programa  
(b) FMT su lagane na račun i primjenjive u praksi  
(c) FMT se najčešće koristi za sintezu programa  
(d) FMT uvijek mogu garantirati da je sustav bez pogrešaka
15. Što od navedenog nije uobičajeno naziv datoteke koju koristi sustav Java PathFinder?
- (a) `ipf.properties`  
(b) `bounded_buffer.jpf`  
(c) `include.properties`  
(d) `site.properties`
16. Booleova funkcija  $F = a + b$  može se operatora ITE kao:
- (a)  $ITE(a, 1, b)$   
(b)  $ITE(a, b, 1)$   
(c)  $ITE(a, b, 0)$   
(d)  $ITE(a, 0, b)$
17. Na modelu prikazanom na slici, za sva stanja, a formula  $AF$  (stvarno) stanja. Stanje  $S_0$  je početno.
- 
- ```
graph LR; S0((S0)) --> S0; S0 --> S1((S1)); S1 --> S2((S2)); S2 --> S0; S2 --> S2;
```
- (a) 1: ne vrijedi, 2: vrijedi
(b) 1: vrijedi, 2: ne vrijedi
(c) 1: ne vrijedi, 2: ne vrijedi
(d) 1: vrijedi, 2: vrijedi
18. SAT-rješavači koriste se u sintezi modela. Ograničenje
- (a) Da se razmatra samo jedna varijabla
(b) Da se provjerava samo jedna varijabla
(c) Da se ograničavaju samo jedna varijabla
(d) Da se razmatra samo jedna varijabla



00:18



20200629_080248.mp4



32. Navedi blok `never` (`never { ... }`) u jeziku Promela:
- (a) Je formula LTL logike implementirana kao Büchi automat u sintaksi jezika Promela
 - (b) Definira Kripke strukturu u jeziku Promela
 - (c) Realizira iznimke (engl. exceptions) u jeziku Promela
 - (d) Nakon naredbe `timeout` definira protuprimjere ako dođe do zastoja (engl. deadlock)
33. U dijelu koda u NuSMV-u:
- ```
next (turn) :=
 case
 turn = myturn & st = c : TRUE;
 TRUE : FALSE;
 esac;
```
- varijabla `turn` postaje istinita (TRUE).
- (a) Ako nije zadovoljeno `turn = myturn & st = c`
  - (b) Samo ako trenutno vrijedi `turn = FALSE`
  - (c) Ako je zadovoljeno `turn = myturn & st = c`
  - (d) Ona je uvijek istinita
34. Zadan je dio Promela procesa:
- ```
1 int n=0;
2 if
3   : n == 0; n++;
4   : n == 1; n--;
5   : n == 2; n++;
6   : n < 0 -> timeout;
7 fi
8 skip;
```
- Odredite vrijednost varijable `n` u redu 8
- (a) `n = 2`
 - (b) `n = 1`
 - (c) dolazi do globalnog zastoja (deadlock) naredba `skip` u redu 8 se ne izvrši, `n =`
 - (d) `n = 3`



00:15

00:18

