IME I PREZIME:_____ Ak. god. 2017/2018

JMBAG:_____

1. domaća zadaća iz Formalnih metoda u oblikovanju sustava

NuSMV

Najprije je potrebno instalirati sustav NuSMV prema uputama u datoteci "NuSMV_upute_2018.pdf"

1. dio

1.1. Prouči primjer mutex_1ex.smv.

1.2. Specificiraj i napiši u CTL notaciji obilježje sigurnosti (engl. *safety property*):
«Dva procesa ne mogu biti istovremeno u kritičnom odsječku.»

Potrebno je napisati dva oblika obilježja:
a) specifikacija da je moguće jedno nepoželjno ponašanje (engl. *refutation*)
b) specifikacija da nema nepoželjnog ponašanja
Nepoželjno ponašanje je u ovom slučaju istovremeno nalaženje u kritičnom odsječku.

1.3. Utvrdi pomoću sustava NuSMV je li ispunjeno navedeno obilježje. Objasni rezultat na temelju koda primjera (ne na temelju ispisa traga).

1.4. Specificiraj i napiši u CTL notaciji obilježje (engl. *liveness property*):
«Ako proces pokuša ući u kritični odsječak, konačno će i ući»
Specifikaciju napiši za oba procesa.

1.5. Utvrdi da li je zadovoljeno navedeno obilježje. Koji su sve problemi s ovom implementacijom?

1.6. U mutex_1ex.smv dodaj ograničenje pravednosti (engl. *fairness*): svaka instanca procesa obavlja se beskonačno mnogo puta.

1.7. Ponovno provjeri prethodno obilježje. Što smo postigli s ovim ograničenjem pravednosti?

1.8. U mutex_1ex.smv još dodaj ograničenje pravednosti: svaka instanca procesa ne može beskonačno dugo ostati u **kritičnom** odsječku. Provjeri sad svojstvo životnosti za mutex_1ex.smv.

1.9. U mutex_1ex.smv dodaj još jedno ograničenje pravednosti: svaka instanca procesa ne može beskonačno dugo ostati u **nekritičnom** odsječku. Provjeri sad svojstvo životnosti za mutex_1ex.smv.

1.10. Specificiraj i napiši u CTL notaciji:

«Ako proces proc0 uđe u kritični odsječak, proc0 neće ponovo ući u kritični odsječak sve dok proc1 nije prošao kroz svoj kritični odsječak.»

1.11. Utvrdi je li zadovoljeno navedeno obilježje za mutex_1ex.smv (uz ograničenja pravednosti). Koja obilježja protokola međusobnog isključivanja rješavaju ograničenja pravednosti prethodno navedena, a koji problem je još uvijek prisutan?

2. dio

2.1. Prouči primjer mutex_2ex.smv

2.2. Specificiraj i napiši u CTL notaciji obilježje sigurnosti (engl. *safety property*):
«Dva procesa ne mogu biti istovremeno u kritičnom odsječku.»

Potrebno je napisati dva oblika obilježja:
a) specifikacija da je moguće jedno nepoželjno ponašanje (engl. *refutation*)
b) specifikacija da nema nepoželjnog ponašanja

2.3. Utvrdi da li je ispunjeno zadano obilježje. Objasni rezultat na temelju koda primjera (ne na temelju ispisa traga).

2.4. Specificiraj i napiši u CTL notaciji obilježje (engl. *liveness property*):
«Ako proces pokuša ući u kritični odsječak, konačno će i ući»
Specifikaciju napiši za oba procesa.

2.5. Utvrdi je li zadovoljeno navedeno obilježje. Objasni koji je problem u ovoj implementaciji međusobnog isključivanja.

2.6. Specificiraj i napiši u CTL notaciji taj problem i provjeri ga pomoću NuSMV sustava.

2.7. Prouči primjer mutex_3ex.smv.

2.8. Je li zadovoljeno obilježje sigurnosti (2.dio, 2. pitanje)?

2.9. Je li zadovoljeno obilježje životnosti (2. dio, 4. pitanje)?

2.10. Dodajte sad ograničenja pravednosti kao kod zadataka 1.6, 1.8 i 1.9. Je li sad zadovoljeno obilježje životnosti?

2.11. Koji je problem u ovoj implementaciji međusobnog isključivanja (bez obzira na uključena ograničenja pravednosti)? Gdje sustav može «zapeti»? Problem specificiraj u CTL notaciji i provjeri pomoću sustava NuSMV.

2.12. Prouči primjer mutex_4ex.smv
Ovo je primjer uspješne implementacije međusobnog isključivanja. Zasniva se na rješenju kojeg je predložio T. Dekker a opisao E. W. Dijkstra.

2.13. Provjeri svojstva sigurnosti i životnosti. Jesu li zadovoljena (uz dodavanje tri ograničenja pristranosti iz 1. dijela)?

2.14. Koje se ideje za kontrolu pristupa kritičnom odsječku iz prethodnih (neuspješnih) pokušaja nameću u ovom rješenju?

2.15. Prouči primjer mutex_5ex.smv
Ovaj je primjer implementacija Petersonovog algoritma, koji predstavlja pojednostavnjenje prethodnog (Dekkerovog) algoritma.

2.16. Je li zadovoljeno obilježje sigurnosti?

2.17. Specificiraj i napiši u CTL notaciji obilježje životnosti. Provjeri ga pomoću sustava NuSMV. Je li to obilježje zadovoljeno (uz dodavanje tri ograničenja pravednosti iz 1. dijela)?

3. dio

Prouči potpoglavlja 3.1, 3.2, 3.5 i 3.7 iz NuSMV priručnika "NuSMV 2.6 User Manual". Nakon toga riješi sljedeće zadatke:

3.1. Pokreni interaktivno ljusku NuSMV-a. Učitaj model zadan datotekom mutex_1ex_int.smv.

3.2. Inicijaliziraj sustav za verifikaciju. Ukratko obrazloži što se sve događa prilikom pokretanja naredbe "go".

3.3. Simuliraj kretanje kroz 3 stanja (od proizvoljno odabranog početnoga). Navedi dvije naredbe koje se koriste da bi se to ostvarilo. Koju naredbu treba koristiti da bi se ispisao trag prolaska kroz ta stanja?

3.4. Provjeri stroj s konačnim brojem stanja. Kakva je relacija prijelaza tog automata? Može li doći do potpunog zastoja?

3.5. Koliko ukupno postoji stanja u modelu, a koliko postoji dosezljivih (engl. *reachable*) stanja? (napomena: *diameter* - promjer FSM-a je minimalan broj koraka potrebnih da bi se došlo do svih dosezljivih stanja)

3.6. Provjeri prvu po redu CTL specifikaciju (redni broj 0). Je li ona istinita ili lažna? Koje obilježje protokola međusobnog isključivanja se njome provjerava? Je li to obilježje zadovoljeno?

3.7. Provjeri drugu po redu CTL specifikaciju (redni broj 1). Je li ona istinita ili lažna? Koje obilježje protokola međusobnog isključivanja se njome provjerava? Je li to obilježje zadovoljeno?

3.8. Sada ukloni obilježja pravednosti iz datoteke mutex_1ex_int.smv, ponovi postupak učitavanja i pripreme za verifikaciju te provjeri drugu po redu CTL specifikaciju. Ima li kakve promjene u odnosu na prethodni zadatak?

3.9. Prouči naredbe za provjeru svojstava sustava za rad u stvarnom vremenu koje su zadane s ključnom riječi "COMPUTE" u datoteci mutex_1ex_int.smv. Koje je značenje svake od tih naredbi?

3.10. Provjeri te naredbe u sustavu NuSMV (prva COMPUTE naredba ima redni broj 2 u modelu, a druga redni broj 3). Navedi rezultat izvođenja tih dviju naredbi. Uzima li naredba COMPUTE u obzir navedena ograničenja pravednosti?

4. dio

4.1. Prouči primjer ferryman.smv.

4.2. Specificiraj i napiši u CTL notaciji obilježje:
«Ne postoji siguran put kojim se dolazi do cilja problema.»
Pritom se u specifikaciji smiju koristiti već definirane makro-instrukcije.

4.3. Provjeri zadano svojstvo. Je li ono zadovoljeno? Što nam u ovom slučaju daje ispis traga programa? Opiši redoslijed izvođenja kojim se uspješno dolazi do cilja problema.

4.4. Zadani kôd u NuSMV-u sadrži implicitni nedeterminizam uzrokovan varijablom *request.* Izmijeni zadani kôd tako da sadrži **isključivo** eksplicitni nedeterminizam.

```
MODULE main
VAR
      request: boolean
      flag: {red, blue};
ASSIGN
      init(flag) := red;
      next(flag) := case
                        request:blue;
                        TRUE: red;
      esac;
```

4.5. Za zadani kôd u NuSMV-u nacrtaj odgovarajuću Kripke strukturu i odredi:
a) skup svih mogućih stanja – S_A
b) skup svih dosezljivih stanja – S_R (uz pretpostavku da su sva početna stanja dosezljiva)

```
MODULE main
VAR
      request : boolean;
      status : {ready, busy};
      negReq : boolean;
ASSIGN
      init(request) := TRUE;
      init(status) := ready;
      next(request) := FALSE;
      next(status) := case
                        request:ready;
                        TRUE: busy;
      esac;
      next(negReq) := !request;
```