

Formális módszerek (VIMIM100)						
ZH1 minta feladatsor						
Név: _____	1.	2.	3.	4.	5.	Σ
NEPTUN kód: _____	12 pont	14 pont	8 pont	8 pont	8 pont	50 pont

### 1. Kiskérdések (12 pont)

1.1. Adja meg, mi az alapvető különbség a *lineáris idejű* és az *elágazó idejű* temporális logikák között! Írjon fel egy-egy példa kifejezést mindkét típusú logika esetén! 3 pont

1.2. Egy tárgyunk állapotai a Neptun rendszerben a következők lehetnek:  $\{FV = Felvett, ASZ = Aláírást szerzett, TT = Teljesített, NT = Nem teljesített\}$ . Fogalmazza meg az alábbi követelményeket CTL\* kifejezések segítségével! 3 pont

a) Ha egyszer aláírást szereztünk a tárgyból, akkor egy sikeres vizsga letételéig *Nem teljesített* lesz a státusza.

b) A tárgy státusza nem lehet egyszerre teljesített és nem teljesített is.

1.3. Ismertesse a tabló módszer PLTL modellellenőrzésre való kiterjesztésének alapötletét az  $X$  és az  $U$  operátor felbontási szabályának bemutatásával! 3 pont

1.4. Mutasson egy-egy példát (ha szükséges, akkor külön lapon) a következőkre: 3 pont

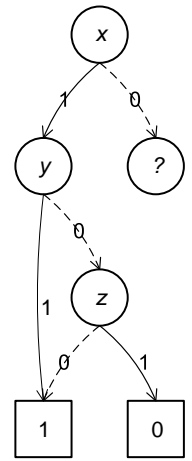
a) döntési fa, ami nem bináris;

b) bináris döntési fa, ami nem rendezett;

c) rendezett bináris döntési fa, ami nem redukált!

## 2. Bináris döntési diagramok (14 pont)

Az  $f$  függvény ROBDD alakban a jobb oldali ábrán látható, azonban az egyik csomópont alatti részek törölődtek. A hiányos csomópont (amit ? jelez az ábrán) által reprezentált logikai függvényt azonban ismerjük:  $\neg y \wedge \neg z$ .



2.1. Írja fel a teljes  $f$  függvényt algebrai alakban (képlettel) majd rajzolja fel (külön lapon) redukált rendezett bináris döntési diagram (ROBDD) alakban! A változók kiértékelésének sorrendje legyen  $x, y, z$ .

5 pont

2.2. A  $g$  függvény az alábbi alakban adott:  $g = (\neg y \wedge \neg z) \vee (\neg x \wedge \neg y)$   
Rajzolja fel a  $g$  függvény ROBDD-jét, a változók kiértékelésének sorrendje legyen  $x, y, z$ .

4 pont

2.3. Legyen az  $m$  függvény a következő:  $m = f \vee g$ .  
Rajzolja fel az  $m$  logikai függvényt ROBDD alakban! Az ROBDD konstruálását közvetlenül az  $f$  és  $g$  ROBDD alakjának felhasználásával végezze el! A változók kiértékelésének sorrendje legyen ez esetben is  $x, y, z$ .

5 pont

## 3. Szemantika alapú modellellenőrzés (8 pont)

Egy zsilipkamra két ajtóval rendelkezik, melyek soha nem lehetnek egyszerre nyitva. Az ajtók állapotát  $x$  és  $y$  bináris állapotváltozók írják le, egy harmadik  $s$  változó pedig azt tárolja, hogy valaki halad-e át a zsilipen. *Alapállapotban* mindkét ajtó zárva van, és senki nincs az ajtók között ( $x=y=s=0$ ). Ha valaki át szeretne haladni, először kinyílik az első ajtó (ez legyen  $x$ ), majd a rendszer észleli a belépő embert, ezért (ezzel egy időben) újra bezáródik az ajtó. Ezután a folyamat a másik ajtóval (ez legyen  $y$ ) megismétlődik: az ajtó kinyílik, majd az ember távozik és az ajtó bezárul, ezzel a rendszer visszatér az alapállapotba.

3.1. Rajzolja fel (külön lapon) a rendszer Kripke struktúra modelljét, jelölje be a rendszerállapotok kódolását, és címkézze meg az állapotokat az *alap*, *ajtónyitva* és *bentvan* címkéket használva!

1 pont

3.2. Adja meg annak az állapothalmaznak a karakterisztikus függvényét, amely azokat az állapotokat tartalmazza, amelyek el vannak látva az *alap* vagy az *ajtónyitva* címkével!

1 pont

3.3. Adja meg annak az állapotátmenet halmaznak a karakterisztikus függvényét, amely az *ajtónyitva* címkével ellátott állapotokból kiinduló állapotátmeneteket tartalmazza!

2 pont

3.4. A tanult iteratív állapotcímkézési eljárást végrehajtva ellenőrizze a modellen, hogy az alapállapotból, mint kezdőállapotból kiindulva teljesül-e az alábbi CTL kifejezés:  $A((ajtónyitva \vee alap) U bentvan)$ ! Az iteráció minden lépéséhez adja meg a címkéző kifejezést és (felsorolással) a címkézett állapotok halmazát!

4 pont

#### 4. LTL követelményformalizálás (8 pont)

Egy doktorandusz hallgató tudományos cikket ír egy nemzetközi konferenciára. A leadási határidőhöz közeledve *hócipőjének* állapotait a jobb oldali ábrán látható Kripke struktúra modellezi.

A *doktorandusz* viselkedését a következő szabályok határozzák meg (ezek <feltétel> -> <állapotváltozás> alakúak):

```
var doktorandusz: {ír, pihen, túrórudizik, dühöng}
initialization:
```

```
    doktorandusz := ír
```

```
transition rules:
```

```
    doktorandusz = ír && hócipő = félig -> doktorandusz := ír
```

```
    doktorandusz = ír && hócipő = félig -> doktorandusz := túrórudizik
```

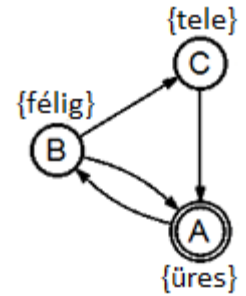
```
    doktorandusz = túrórudizik && hócipő = üres -> doktorandusz := ír
```

```
    doktorandusz = ír && hócipő = tele -> doktorandusz := dühöng
```

```
    doktorandusz = dühöng -> doktorandusz := pihen
```

```
    doktorandusz = pihen && hócipő = üres -> doktorandusz := ír
```

```
    doktorandusz = pihen && hócipő = félig -> doktorandusz := ír
```



4.1. Rajzolja fel azt a Kripke struktúrát, ami a *doktorandusz* viselkedését modellezi! Használja az {ír, pihen, túrórudizik, dühöng} állapotcímkéket!

2 pont

4.2. Formalizálja LTL kifejezések segítségével az alábbi követelményeket, amelyek a *doktorandusz* viselkedésére minden esetben vonatkoznak!

6 pont

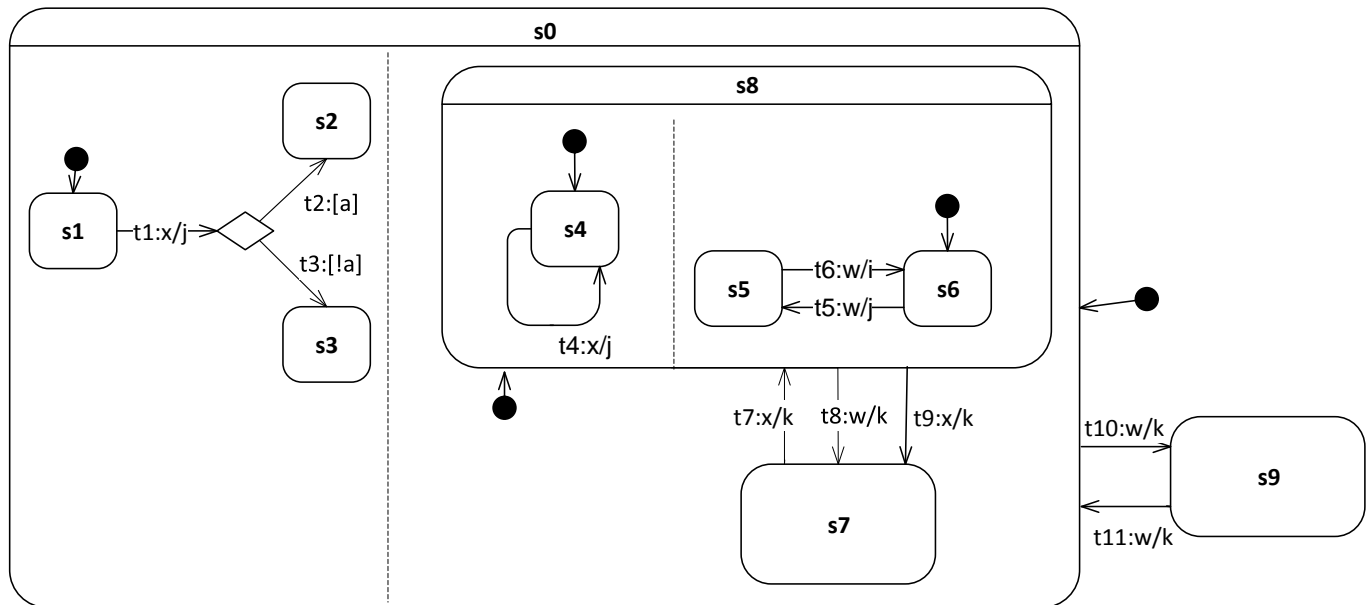
4.2-1. Ha tele van a hócipője, akkor dühöng, majd a következő időpillanatban pihen.

4.2-2. A doktorandusz előbb-utóbb folytatja az írást.

4.2-3. Ha a doktorandusz túrórudizik, akkor addig nem pihen, amíg nincs tele a hócipője.

## 5. UML állapottérképek (8 pont)

Tekintsük az alábbi állapottérképet, melyben minden  $s_k$  állapothoz tartoznak  $s_k.entry$  és  $s_k.exit$  akciók is, melyek az adott állapot belépési és a kilépési akcióit jelentik! A nyilakra írt kifejezések átmenetnév: trigger [őrfeltétel] / akció alakúak.



A rendszer kezdőállapotban van, az  $a$  boolean változó értéke  $false$ . Az esemény-sorrendezőtől egy „ $x$ ” esemény érkezik.

5.1. Melyek az engedélyezett állapotátmenetek?

1 pont

5.2. Mely engedélyezett állapotátmenetek állnak egymással konfliktusban?

1 pont

5.3. Hogy néz ki a tüzelhető állapotátmenetek halmaza (konfliktusfeloldás után)?

1 pont

5.4. Hogy néz(nek) ki a következő stabil állapotkonfiguráció(k)? (Ha több is lehetséges, mindet adja meg!) Adja meg azt is, hogy milyen akciók hajtódnak végre tüzelés közben és milyen sorrendben!

2 pont

5.5. Az esemény-sorrendezőtől ezután egy „ $w$ ” esemény érkezik. Adja meg a tüzelhető állapotátmenetek halmazát (ha szükséges, akkor konfliktusfeloldás után) és a következő stabil állapotkonfigurációt! Ha az előző pontban több állapotkonfiguráció adódott, akkor mindegyikre adja meg ezeket!

3 pont