

Gyakorló feladatok a zárthelyi dolgozathoz (GTK levelező képzés)

1. Követelmények formalizálása temporális logika alkalmazásával

Egy vasúti kereszteződést biztosító fénysorompó viselkedését az állapotaihoz rendelt következő atomi kijelentésekkel jellemezzük: $\{kikapcsolt, fehér, piros\}$

A kereszteződéshez érkező autós viselkedését az állapotaihoz rendelt következő atomi kijelentésekkel jellemezzük: $\{érkezik, körülnéz, megáll, áthalad\}$

Formalizálja LTL kifejezések segítségével az alábbi követelményeket, amelyek az autós viselkedésére minden esetben vonatkoznak:

- Kikapcsolt állapotú fénysorompó esetén az autós körülnéz és a következő időpillanatban vagy áthalad, vagy megáll.
- Az autós előbb-utóbb át fog haladni a vasúti kereszteződésen.
- Ha egy autós érkezésekor piros a lámpa, akkor az autós addig nem halad át, amíg fehérre nem vált a fénysorompó.

2. Követelmények formalizálása temporális logika alkalmazásával

Egy bonyolult szimulációt futtató szerver állapotait a következő atomi kijelentésekkel jellemezzük: $\{kikapcsolt, várakozó, bemelegítés, szimuláció\}$

A szerverszoba hűtőberendezésének működését az állapotaihoz rendelt következő atomi kijelentésekkel jellemezzük: $\{készlet, normál, maximális\}$

Formalizálja LTL kifejezések segítségével az alábbi követelményeket, amelyek a rendszer működésére minden esetben vonatkoznak:

- Ha egy adott pillanatban a szimuláció a hűtőberendezés készlet állapota mellett zajlik, akkor a következő pillanatban a szerver várakozó állapotra kapcsol.
- Előbb-utóbb elkezdhető a szimuláció.
- Csak úgy hajtható végre szimuláció, ha volt bemelegítés a hűtőberendezés normál működése mellett.

3. Modellezés és modellellenőrzés

Egy informatikus hallgató „állapotait” aszerint különböztetjük meg, hogy éppen kávézik vagy nem, valamint éppen alszik vagy nem. A hallgató három tevékenységét különböztetjük meg: tanulás közben kávézik és nem alszik; ezután vizsgázik, ahol nem kávézik és nem is alszik; a vizsgázás után pihen, ekkor alszik és nem kávézik. A hallgató alapállapota a tanulás, amit a vizsgázásig nem is hagy abba. Tanulás nélkül a hallgató nem vizsgázik; a vizsgázás után közvetlenül pedig nem tanul (csak pihenés után).

- Rajzolja fel a hallgató itt leírt viselkedését modellező Kripke-struktúrát a hallgató kávézását és alvását figyelembe véve! Az egyes állapotokat jellemezze a következő atomi kijelentésekkel: $\{pihen, tanul, vizsgázik\}$

- Ellenőrizze a modellen, hogy a hallgató alapállapotából (ami a tanulás) kiindulva teljesül-e a következő CTL kifejezés: $E(\neg vizsgázik U pihen)$! Válaszát indokolja meg!

4. Modellezés és modellellenőrzés

Egy informatikai rendszer két szerverből áll, ezek egy adatbázisszerver és egy alkalmazáserver, amelyek kikapcsolt vagy bekapcsolt állapotban lehetnek. A szervereket hibamentes esetben egyszerre kapcsolják ki és be. Alaphelyzetben mindkét szerver ki van kapcsolva. A megfelelő üzemi állapot az, amikor mindkét szerver be van kapcsolva. Ha az üzemi állapotban az adatbázisszervert hiba következtében kikapcsolják, az rendszer szinten üzemképtelen állapotnak tekinthető. Ezután az alkalmazáservert is kikapcsolják, majd mindkét szerver bekapcsolásával indítják újra a rendszert.

- Rajzolja fel a rendszer itt leírt működését modellező Kripke-struktúrát az egyes szerverek bekapcsolását és kikapcsolását figyelembe véve! Az egyes állapotokat jellemezze a következő atomi kijelentésekkel: $\{alaphelyzet, üzemi állapot, üzemképtelen\}$
- Ellenőrizze a modellen, hogy az üzemi állapotot kezdőállapotnak tekintve teljesül-e a következő CTL kifejezés: $E(\neg üzemképtelen U alaphelyzet)$! Válaszát indokolja meg!

5. Temporális logikai követelmények értelmezése

Indokolja meg, hogy következő ekvivalencia helyes-e: $F Stop \vee F Start \equiv F (Stop \vee Start)$, ahol \vee a logikai VAGY operátort jelöli, a *Stop* és *Start* pedig atomi kijelentések.

6. Temporális logikai követelmények értelmezése

Indokolja meg, hogy következő ekvivalencia helyes-e: $AG Stop \equiv not EF (not Stop)$, ahol *not* a logikai negálás operátort jelöli, a *Stop* pedig egy atomi kijelentés.

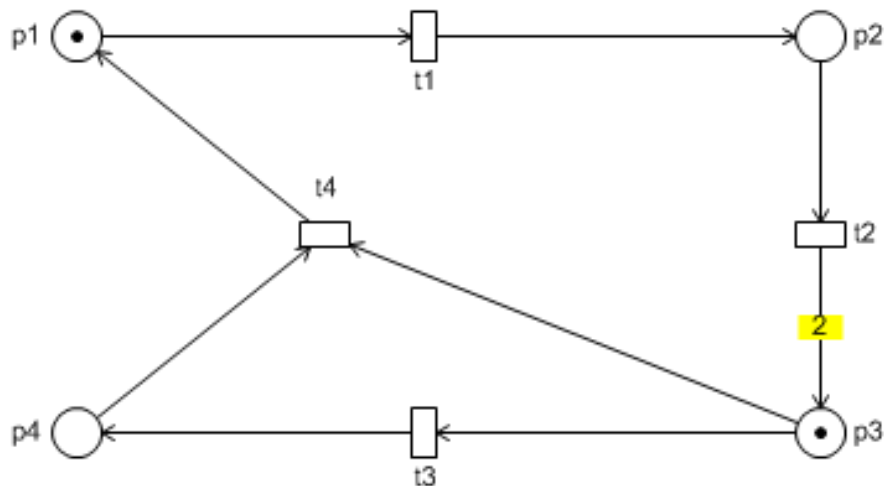
7. Modellezés Petri hálóval

Egy kőbalta készítéséhez egy kődarabra, egy fanyélre, egy kőfúróra és egy kalapácsra van szükség. Kezdetben egy-egy eszköz (kőfúró és kalapács) valamint öt kődarab és öt fanyél áll rendelkezésre. Először a kődarabra kell lyukat fúrni a kőfúró eszközzel, majd a kalapáccsal bele kell verni a fanyelet a lyukba. A kész kőbaltákat egy tárolóba kell tenni. Az eszközöket használat után vissza kell tenni a helyükre, hogy a következő kőbalta készítéséhez használhatók legyenek.

- Készítsen egy Petri hálót, ami modellezi azt a munkafolyamatot, amelynek során Frédi kőbaltákat készít! Tételezzük fel, hogy Frédi csak akkor fog a következő kőbalta készítésébe, ha az előző teljesen készen van.
- Egészítse ki a modellt a következők szerint: Béni is dolgozik, azaz a szabad eszközöket használva a hozzávalókból ő is kőbaltákat készít. Béni is csak akkor fog a következő kőbalta készítésébe, ha az előző teljesen készen van.

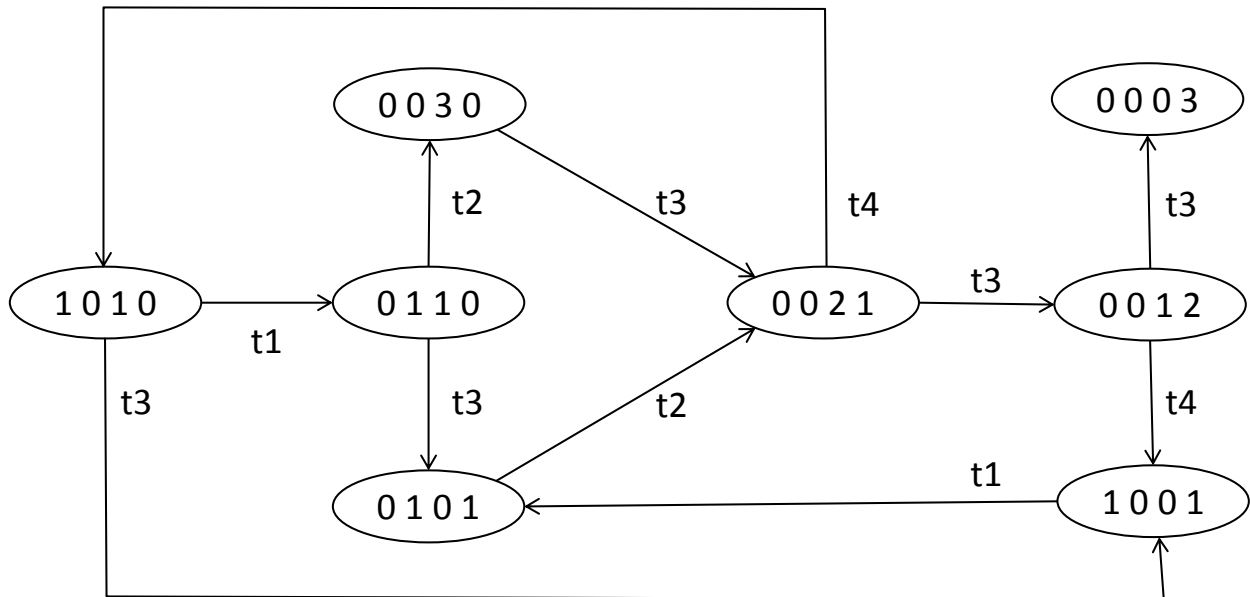
8. Petri hálók elérhetőségi gráfja

Rajzolja fel az alábbi Petri háló elérhetőségi gráfját!



9. Petri hálók dinamikus tulajdonságai

Alább látható egy Petri háló elérhetőségi gráfja. A kezdőállapot az (1 0 1 0) állapot.



Döntse el az alábbi állításokról, hogy igazak vagy hamisak, és indokolja meg döntését!

1. A Petri háló holtpontmentes.
2. A Petri háló korlátos.
3. A Petri háló élő.
4. A Petri háló megfordítható.
5. A (0 0 1 2) állapot visszatérő állapot.
6. A Petri hálónak van visszatérő állapota.
7. A t3 tranzíció L1-élő.
8. A t3 tranzíció L3-élő.
9. A t1, t2, t3, t4 tüzelési szekvencia egy T-invariánst alkot.
10. Az (1, 1, 1, 1) súlyvektor egy P-invariánst határoz meg.