Formális módszerek BMEVIMIMA26

Második zárthelyi: Gyakorló feladatok megoldással

1. Szoftver-modellellenőrzés absztrakcióval

Adott a jobb oldali programrészlet.

```
     z : int
0:   while (z <= 1) {
1:         z := z+1;
     }
2:   assert(z>1);
```

A. Rajzolja le a programrészlethez tartozó *Control Flow Automaton* (CFA) modellt! A vezérlési helyeket a programsorokhoz írt sorszámokkal (0, 1, 2) azonosítsa. Az assertion megsértése esetére vegyen fel egy *err* címkéjű, a jó végállapothoz pedig egy *end* címkéjű vezérlési helyet.

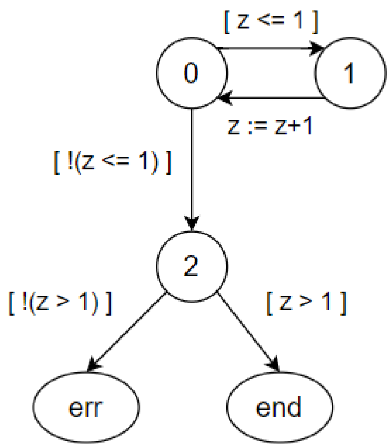
B. A CFA modellellenőrzésére vezérlési hely és predikátum absztrakciót alkalmazunk, ez utóbbihoz egyetlen *(z>=1)* predikátumot használunk. Mik lehetnek az absztrakt állapottérben a kezdőállapotok *(vezérlési hely, predikátumérték)* alakban megadva, ha a program indulásakor a *z* egész értékű változó tetszőleges lehet?

C. Hamis útvonalnak tekinthető-e az *err* vezérlési hely eléréséhez az absztrakt állapottérben lévő alábbi útvonal? Válaszát indokolja!

(0, true) -> (1, true) -> (0, true) -> (2, true) -> (err, true)

Megoldás:

A. A CFA modell:



B. (0, false) és (0, true)

C. Az útvonal hamis, mivel lépésenként vizsgálva:

(0, z>=1) -> (1, z>=1): ez esetben z<=1 (a feltétel miatt) és z>=1 (a predikátum miatt); ez lehetséges, az egyetlen kielégítő eset z=1.

(1, z>=1) -> (0, z>=1): az előző átmenet csak z=1 esettel volt kielégíthető, tehát ez után z=2.

(0, z>=1) -> (2, z>=1): ezen átmenet feltétele !(z<=1), ez konzisztens az eddigiekkel.

(2, z>=1) -> (err, z>=1): ennek feltétele !(z>1), ez viszont ellentmond az útvonal eddigi feltételeinek, amelyek csak úgy voltak kielégíthetőek, ha ebben a lépésben már z=2.

Tehát az útvonal nem konkretizálható.

2. Modellezés Petri-hálóval

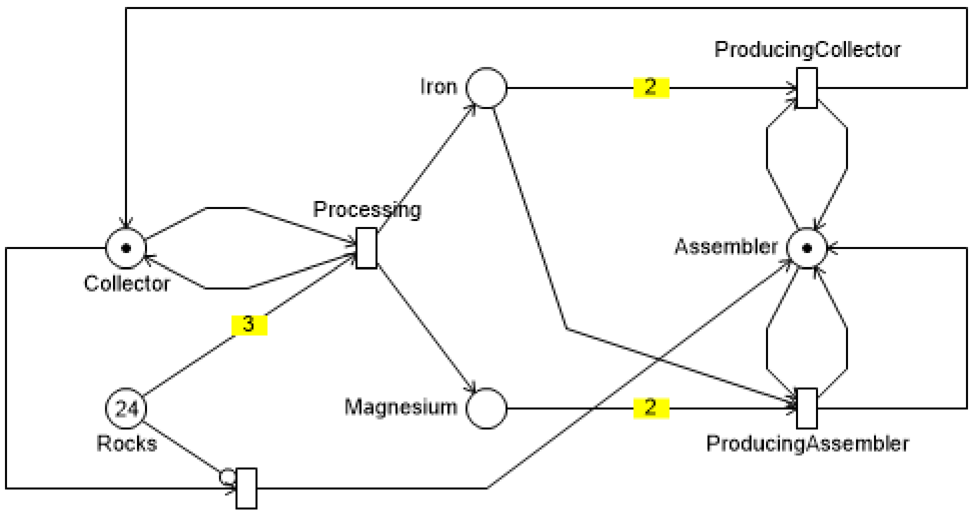
Készítsen egy Petri hálót, ami modellezi az alább leírt folyamatot. Használhatja a Petri-hálók kiterjesztéseit (kapacitáskorlát, tiltó élek, prioritások) is.

Önmagukat replikáló, Mars-felderítő robotok életét szeretnénk modellezni.

- Kezdetben egy *gyűjtögető* és egy *összeszerelő* robot található a Mars felszínén.
- Egy gyűjtögető robot marsfelszíni *kőzeteket* dolgoz fel, melyekből 24 egység van elérhető távolságon belül. Három marsi kőzetegység feldolgozása során pontosan 1 egység *vas* és 1 egység *magnézium* nyersanyag keletkezik. Ha elfogytak a kőzetek, akkor a gyűjtögető robotok összeszerelő robotnak minősítik át magukat.
- Egy összeszerelő robot új gyűjtögető vagy összeszerelő robotokat készít. *Gyűjtögető* robot készítéséhez 2 egység vas szükséges, *összeszerelő* robot készítéséhez pedig 1 egység vas és 2 egység magnézium. Ha mindkét fajta robot készítéséhez van elegendő nyersanyag, akkor véletlenszerűen dönt. Az elkészült robotok azonnal munkába állnak.

Megoldás:

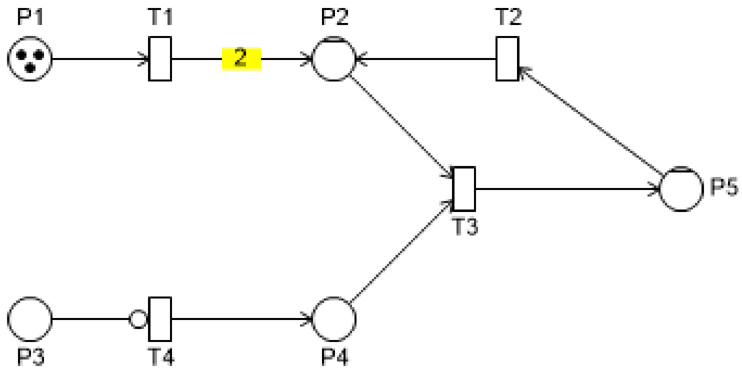
A folyamat Petri-háló modellje:



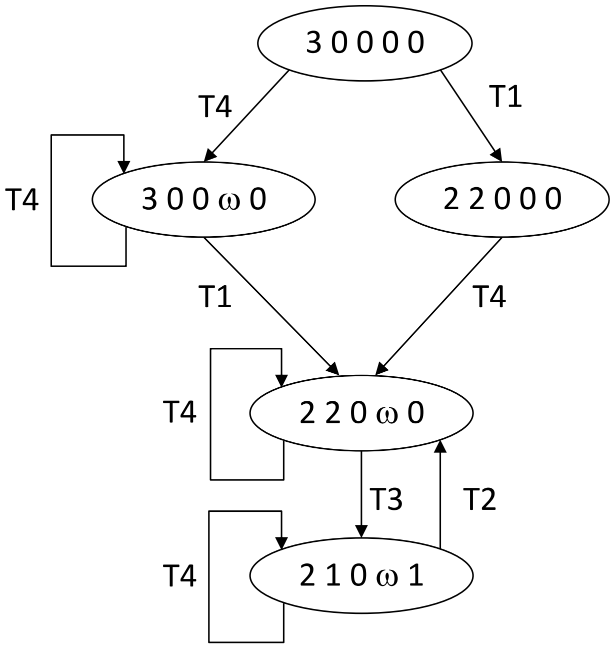
3. Petri-háló állapotterének felvétele

Adott az alábbi Petri-háló, amelyben a P2 és P5 hely kapacitáskorlátosak: K(P2) = 2 és K(P5) = 1.
Az összes további hely végtelen kapacitású. Az élekre írt számok az élsúlyokat jelölik.

Készítse el a Petri-háló *fedési gráfját*. Címkézze fel a fedési gráfban az egyes éleket a tüzelő tranzícióval.



Megoldás:



4. Petri-háló dinamikus tulajdonságai

Az alábbi ábra egy Petri-háló állapotterét mutatja be elérhetőségi gráf alakban. A hálóban 6 tranzíció található, amelyeket t1, ..., t6 címkékkel jelöltünk. Az állapotokba a tokeneloszlás-vektorokat írtuk be, tehát (0 2 0 0) jelentése: m(p1) = 0, m(p2) = 2, m(p3) = 0 és m(p4) = 0. A kezdőállapot a sötét hátterű (2 0 0 0) állapot.

Vizsgálja meg az ábrát, és az alapján adja meg, hogy az adott tulajdonság igaz (I), hamis (H), vagy az elérhetőségi gráf alapján ez nem dönthető el (ND)!

A. A *t5* tranzíció perzisztens.

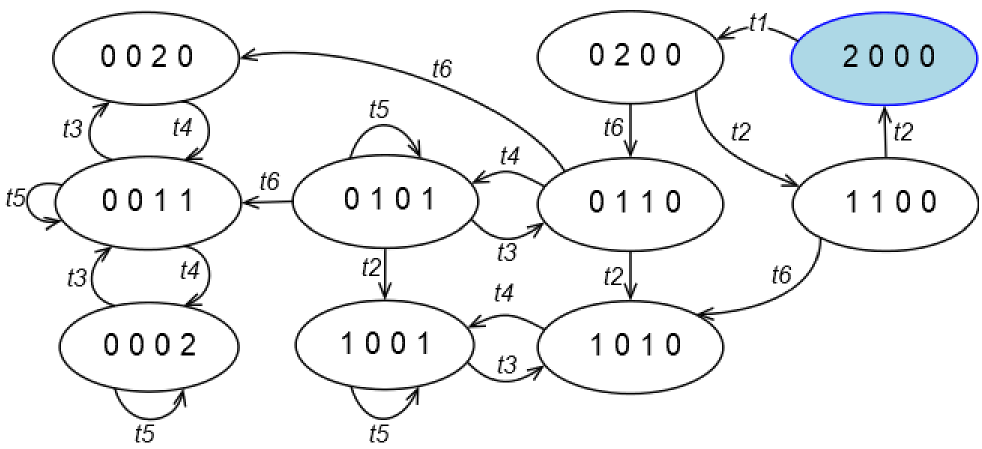
B. A *t2* tranzíció L2-élő.

C. A *t6* tranzíció L4-élő.

D. A háló megfordítható.

E. A hálónak van visszatérő állapota.

F. A háló nem holtpontmentes.



<u>Megoldás:</u>

A. A *t5* tranzíció perzisztens: H, lásd például az (1 0 0 1) állapot esetét: t5 engedélyezett, de ha t3 tüzel helyette, akkor már nem marad engedélyezett.

B. A *t2* tranzíció L2-élő: I, lásd például a (2 0 0 0) – (0 2 0 0) – (1 1 0 0) ciklust, ebből t6 és t6 tüzelésével átlépve a (0 0 2 0) állapotba, t2 már nem tüzelhet a későbbiekben.

C. A *t6* tranzíció L4-élő: H, van olyan állapot, pl. (0 0 2 0), amelyet követően nem tüzelhet.

D. A háló megfordítható: H, lásd például a (0 0 2 0) állapotot: ez elérhető a kezdőállapotból, de innen nem lehetséges a kezdőállapotba visszatérni.

E. A hálónak van visszatérő állapota: I, például a (0 0 2 0) állapot ilyen: az ebből elérhető állapotokból vissza lehet ide térni.

F. A háló nem holtpontmentes: H, mert holtpontmentes.

5. Petri-hálók strukturális tulajdonságai

A. Általános kérdés: Egy gyártási folyamatot reprezentáló Petri-hálóban milyen információhoz juthatunk, ha a T-invariánsait vizsgáljuk?

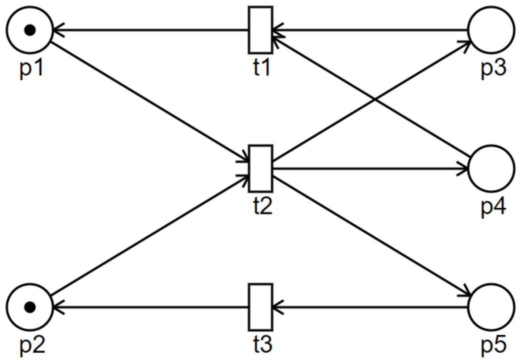
B. Írja fel a lenti ábrán megadott Petri-háló súlyozott szomszédossági mátrixát!

C. Vizsgálja meg, hogy ennek a Petri-hálónak T-invariánsa-e a következő tüzelési vektor. Válaszát indokolja!

$(2,1,3)^T$ oszlopvektor

D. Igaz-e erre a Petri-hálóra az adott kezdőállapot mellett a következő CTL kifejezés, ahol m(pi) a pi (i=1, 2, …, 5) hely jelölését jelenti? Válaszát indokolja!

EF (m(p2) + m(p5) == 2)



Megoldás:

A. Amennyiben a folyamat egészének/részének lépéseit modellező tranzíciók T-invariánst alkotnak, akkor azon lépések ciklikusan végrehajthatók.

B. A súlyozott szomszédossági mátrix (sorokban a tranzíciók, oszlopokban a helyek):

$$\boldsymbol{W} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{-1} & \mathbf{-1} & \mathbf{0} \\ \mathbf{-1} & \mathbf{-1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{-1} \end{bmatrix}$$

C. Az ellenőrzéshez: $\boldsymbol{W^T} * (\mathbf{2}, \mathbf{1}, \mathbf{3})^{\boldsymbol{T}} = (\mathbf{1}, \mathbf{2}, \mathbf{-1}, \mathbf{-1}, \mathbf{-2})$

Az eredmény nem $\underline{\mathbf{0}}$, ezért a megadott vektor a hálónak nem T-invariánsa.

D. Nem igaz.

A kezdőállapotban a tokenek CTL kifejezésben szereplő összege:
$(\mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}) * (\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1})^{\boldsymbol{T}} = \mathbf{1}.$
Továbbá $\boldsymbol{W} * (\mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1})^{\boldsymbol{T}} = (\mathbf{0}, \mathbf{0}, \mathbf{0})$, azaz a CTL kifejezésben szereplő súlyvektor a háló P-invariánsa. Ebből következik, hogy a súlyozott tokenösszeg értéke nem változhat meg 1-ről, azaz nem lehet 2.

6. Színezett Petri-háló

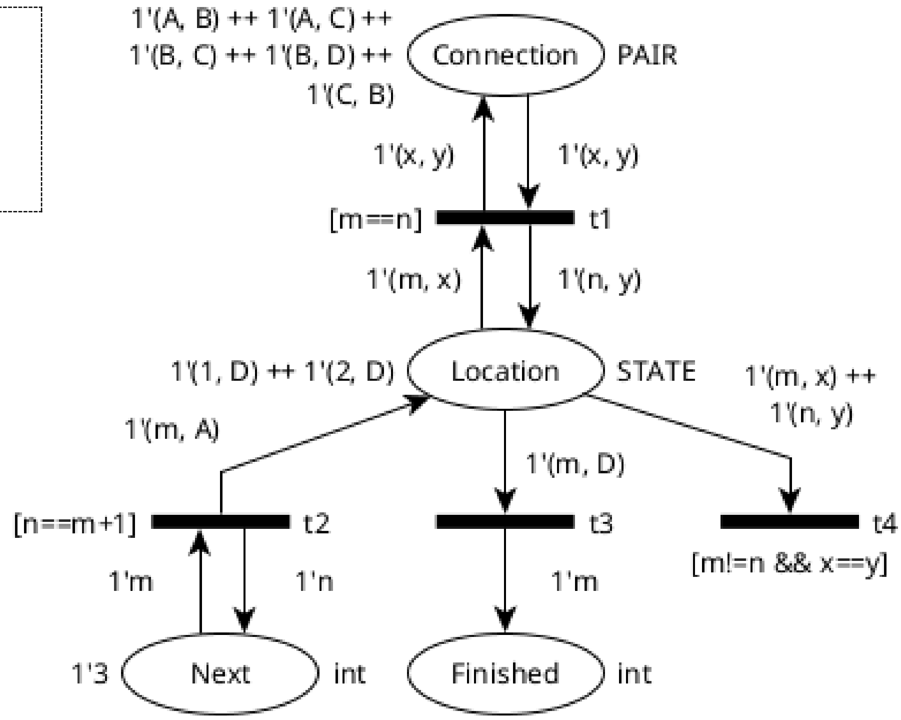
Adott a lenti ábrán látható színezett Petri-háló modell, valamint a hozzá tartozó definíciós mező. A helyek színhalmazai nagybetűsek, az aktuális jelölések a helyek mellé vannak írva, az őrfeltételek szögletes zárójelek között szerepelnek.

A. Sorolja fel, hogy mely tranzíciók és milyen lekötéssel engedélyezettek a háló adott állapotában.

B. Válasszon ki ezek közül *egy* engedélyezett tranzíciót, és adja meg, hogy ennek tüzelése után mi lesz a háló következő jelölése!

C. Elérhető-e a hálóban az adott állapotból holtpont (olyan állapot, ahol egy tranzíció sem tüzelhető)? Válaszát indokolja!

```
colset LOC = with A | B | C | D;
colset STATE = product int * LOC;
colset PAIR = product LOC * LOC;
var x, y: LOC;
var m, n: int;
```



Megoldás:

A. Engedélyezett tranzíció és lekötés	B. Következő jelölés, ha a választott tranzíció tüzel az adott lekötésben			
	Connection	Location	Next	Finished
t2 (m=3, n=4)	1’(A, B) ++ 1’(A, C) ++ 1’(B, C) ++ 1’(B, D) ++ 1’(C, B)	1’(1, D) ++1’(2, D) ++ 1’(3, A)	1’4	-
t3 (m=1)	1’(A, B) ++ 1’(A, C) ++ 1’(B, C) ++ 1’(B, D) ++ 1’(C, B)	1’(2, D)	1’3	1’1
t3 (m=2)	1’(A, B) ++ 1’(A, C) ++ 1’(B, C) ++ 1’(B, D) ++ 1’(C, B)	1’(1, D)	1’3	1’2
t4 (m=1, n=2, x=D, y=D)	1’(A, B) ++ 1’(A, C) ++ 1’(B, C) ++ 1’(B, D) ++ 1’(C, B)	-	1’3	-
t4 (m=2, n=1, x=D, y=D)	1’(A, B) ++ 1’(A, C) ++ 1’(B, C) ++ 1’(B, D) ++ 1’(C, B)	-	1’3	-

C. Nem érhető el holtpont, mivel a t2 tranzíció mindig engedélyezett marad.