Valós idejű tesztek generálása időzített viselkedésmodellekből

Szkupien Péter

Budapesti Műszaki és Gazdaságtudományi Egyetem
Kritikus Rendszerek Kutatócsoport

Bevezetés

- Biztonságkritikus rendszerek → követelmények ellenőrzése
- Modellellenőrzés → teljes állapottér feltárása → modellalapú tesztgenerálás
- Időzített rendszerek → **absztrakt tesztesetek konkretizálása**

Bevezetés

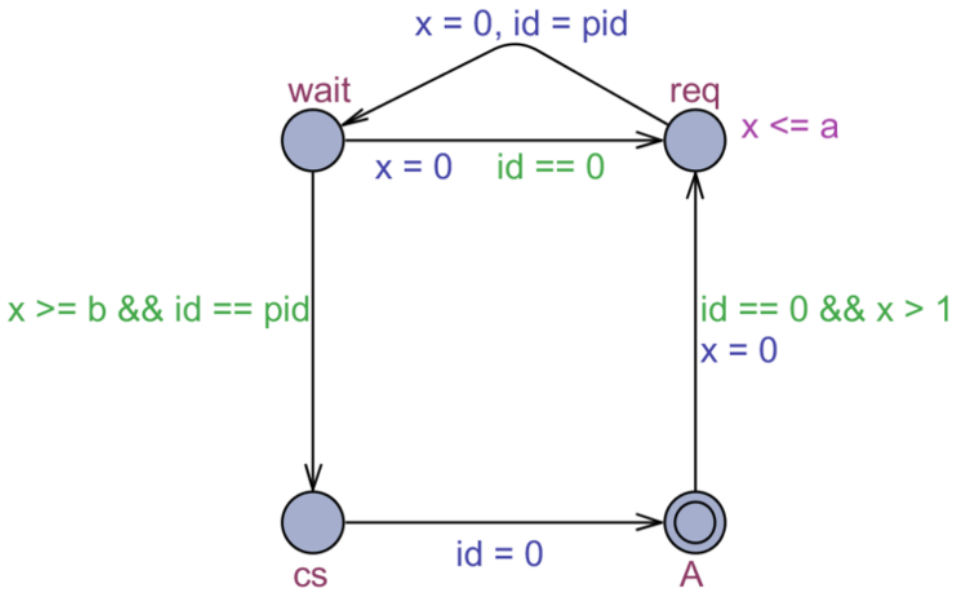
- Biztonságkritikus rendszerek → követelmények ellenőrzése
- Modellellenőrzés → teljes állapottér feltárása → modellalapú tesztgenerálás
- Időzített rendszerek → **absztrakt tesztesetek konkretizálása**

- UPPAAL

- Theta

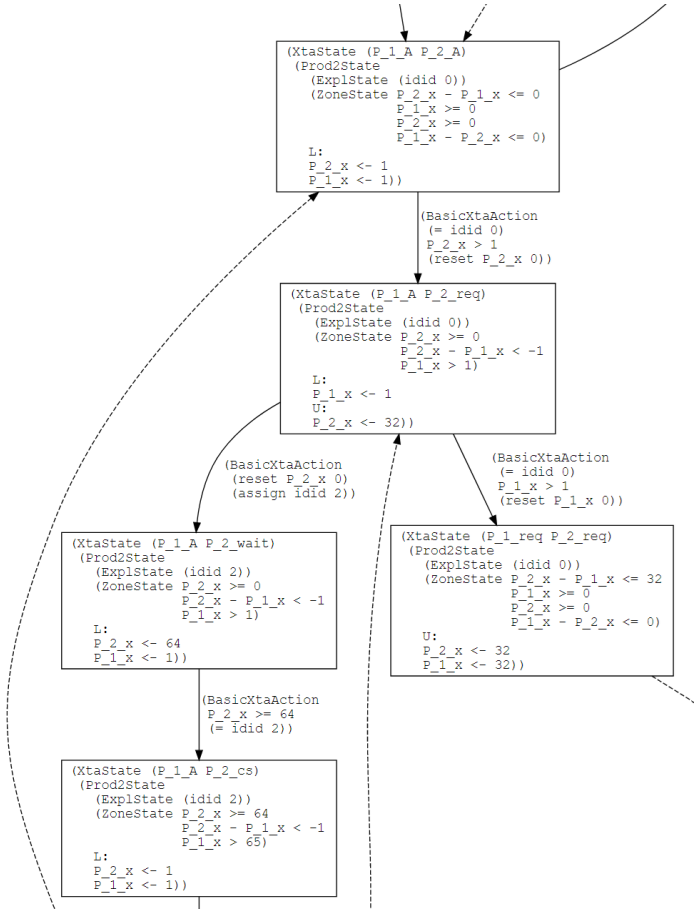
A teljes tesztgenerálási folyamat

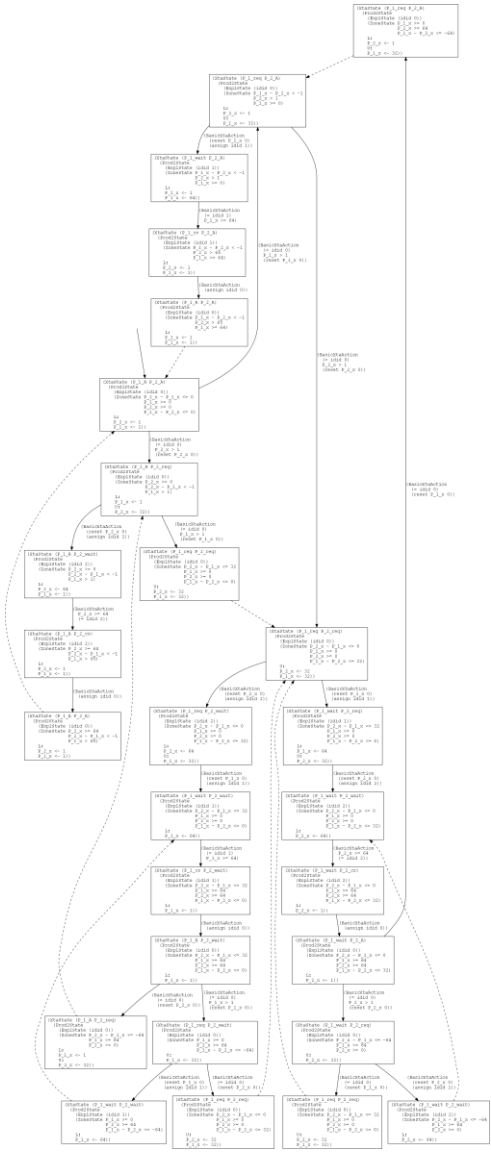
1. Bemenet: időzített automaták hálózata (UPPAAL)



A teljes tesztgenerálási folyamat

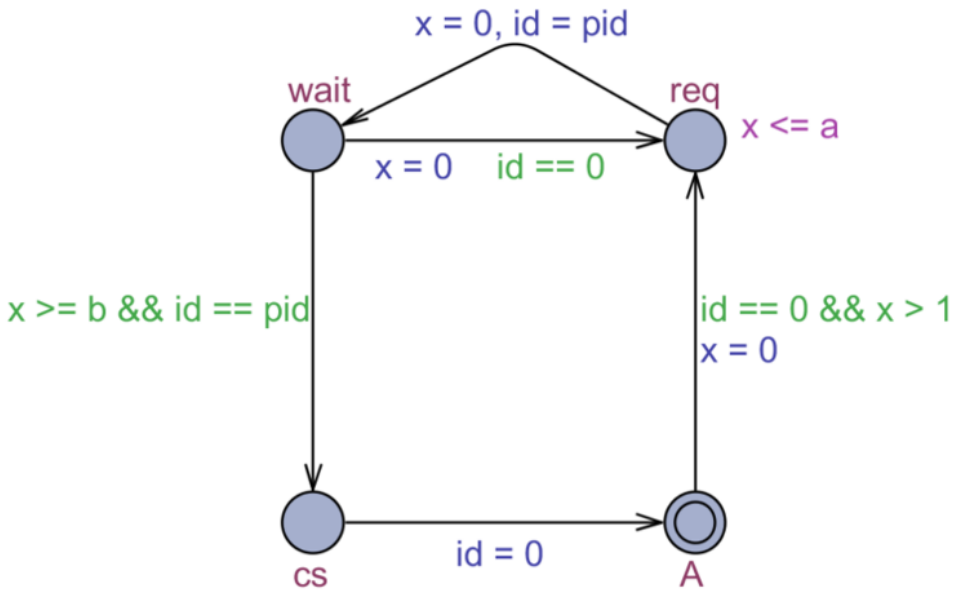
1. Bemenet: időzített automaták hálózata (UPPAAL)
2. Az automatahálózat absztrakt reprezentációja (Theta)

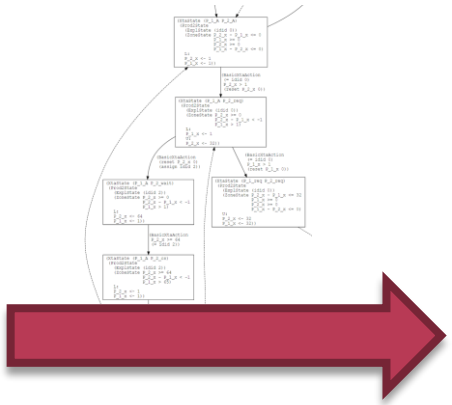




A teljes tesztgenerálási folyamat

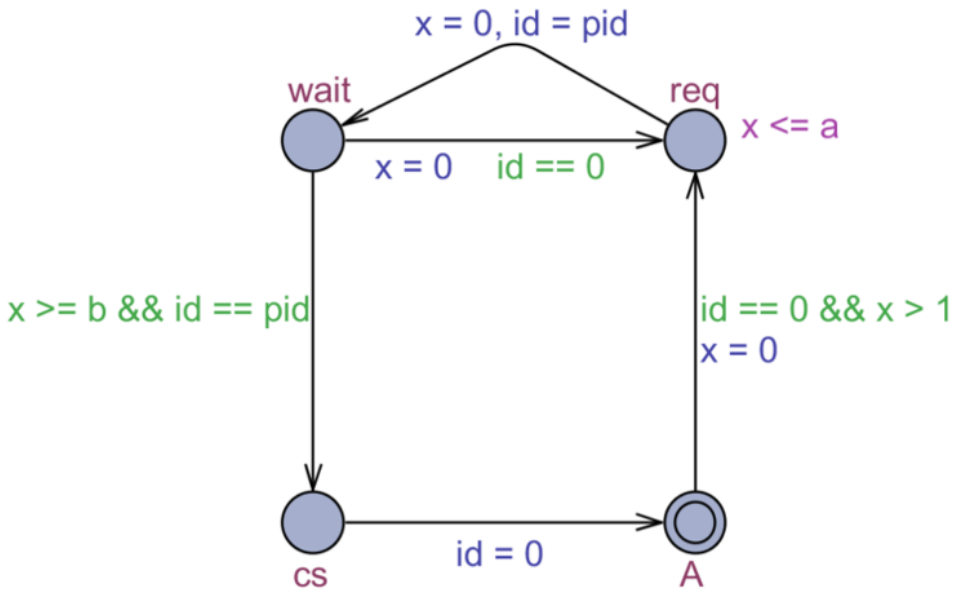
1. Bemenet: időzített automaták hálózata (UPPAAL)
2. Az automatahálózat absztrakt reprezentációja (Theta)
3. **Tesztgenerálás**

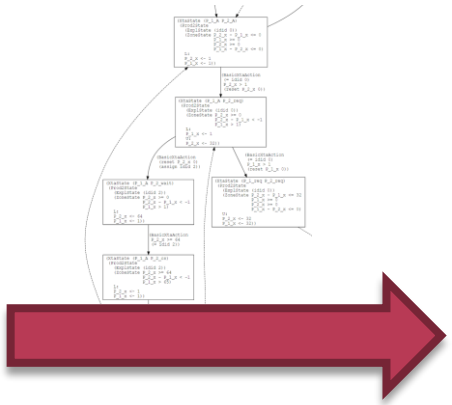


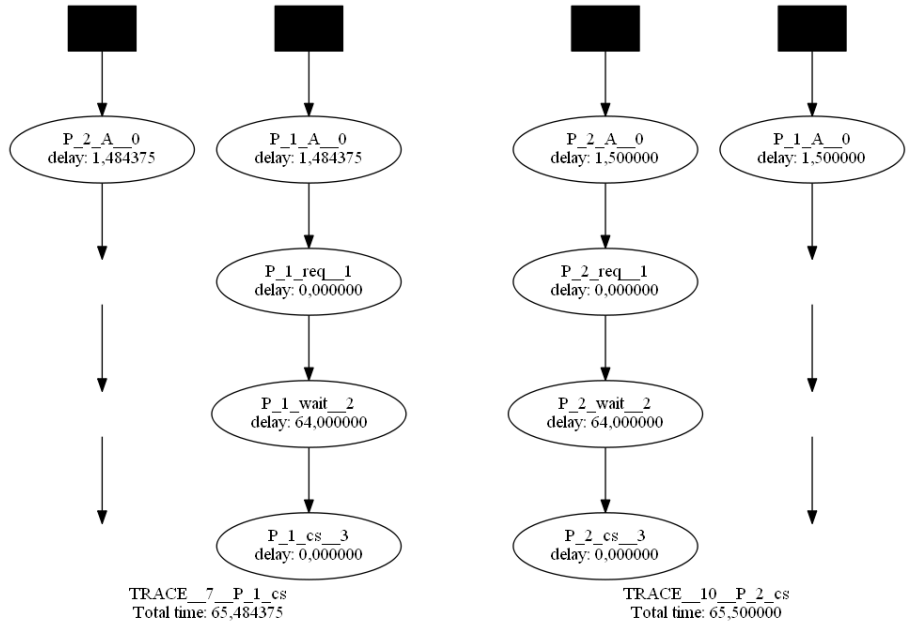


A teljes tesztgenerálási folyamat

1. Bemenet: időzített automaták hálózata (UPPAAL)
2. Az automatahálózat absztrakt reprezentációja (Theta)
3. **Tesztgenerálás**
4. Kimenet: elvárásoknak megfelelő tesztkészlet

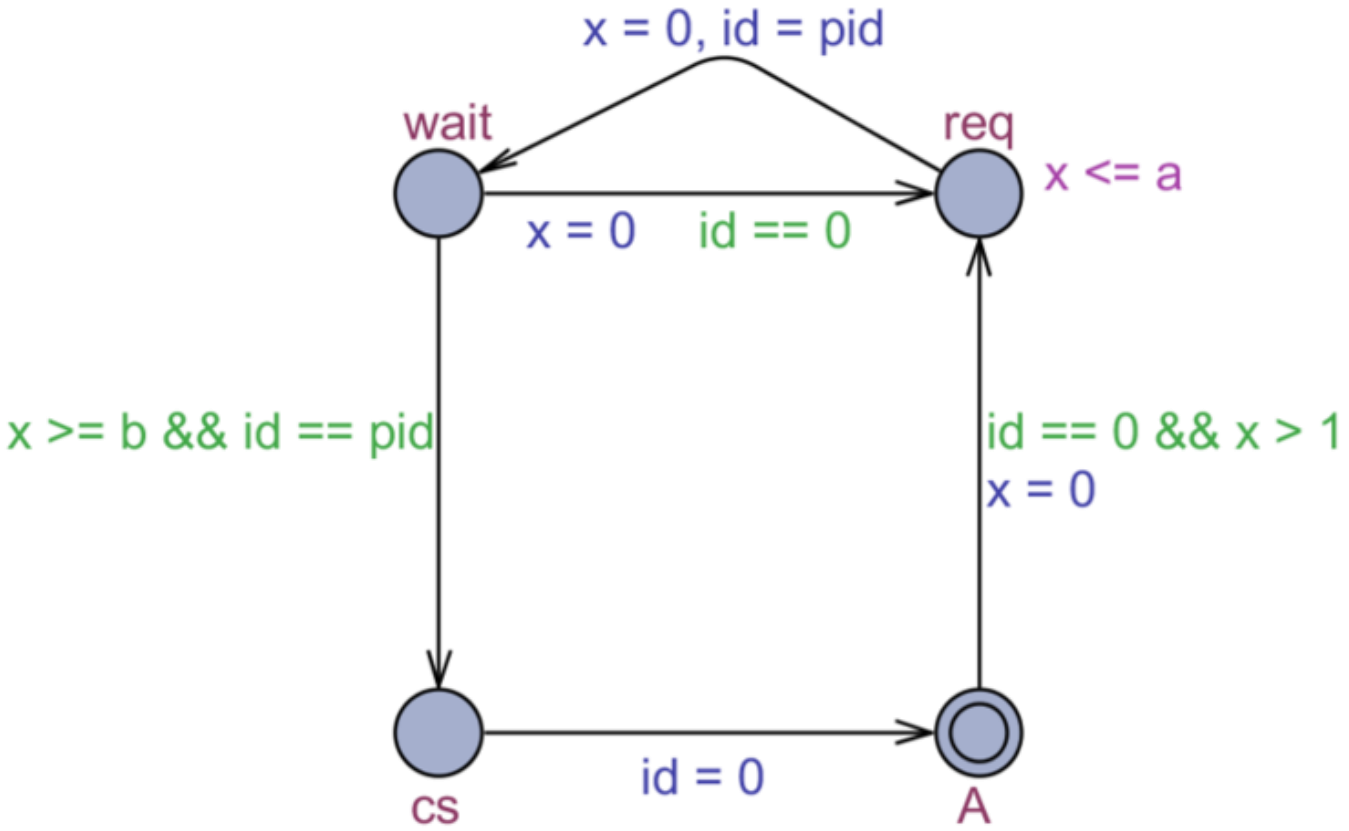






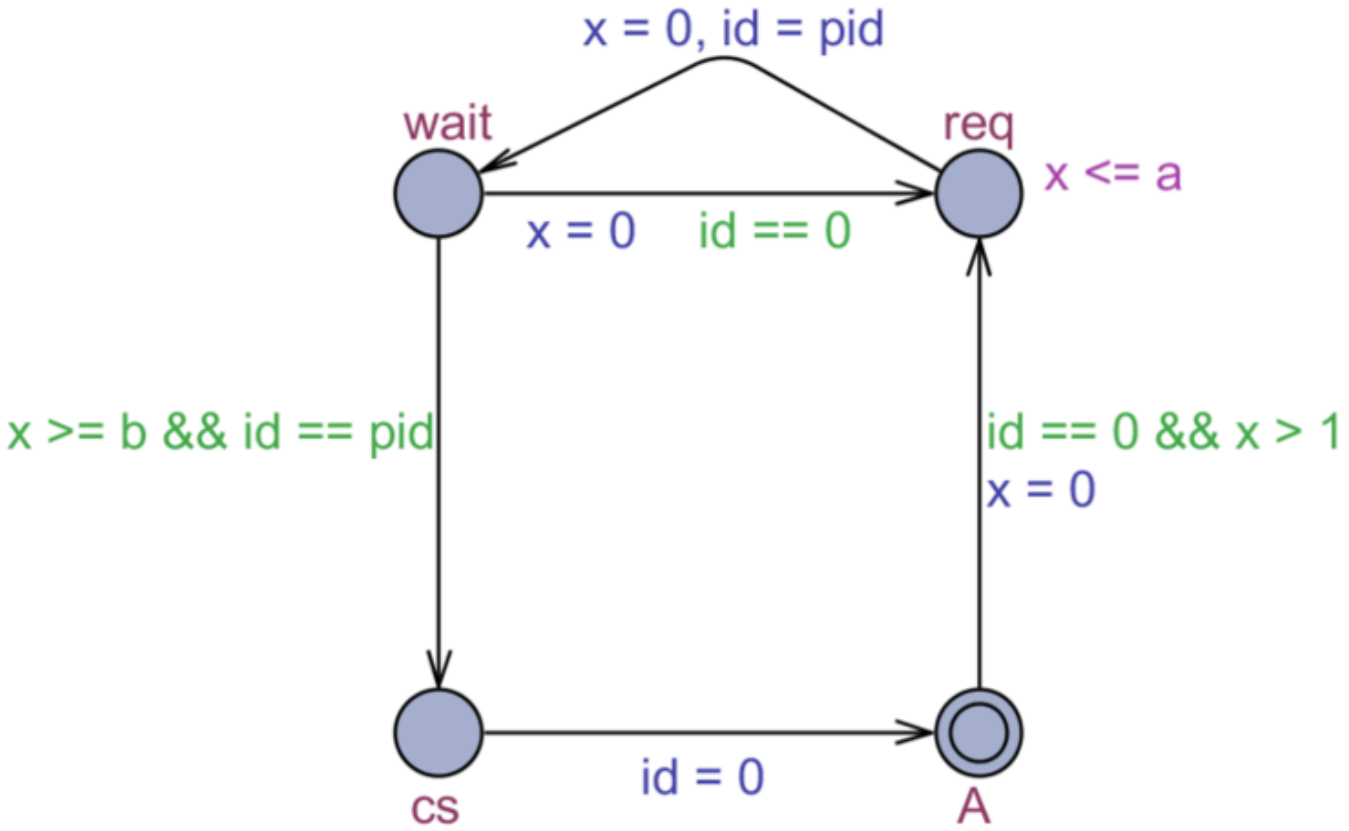
Időzített automaták hálózata

- Időzített automata: véges automata + óraváltozók
  - Állapot: aktív vezérlési hely + óraváltozók állása



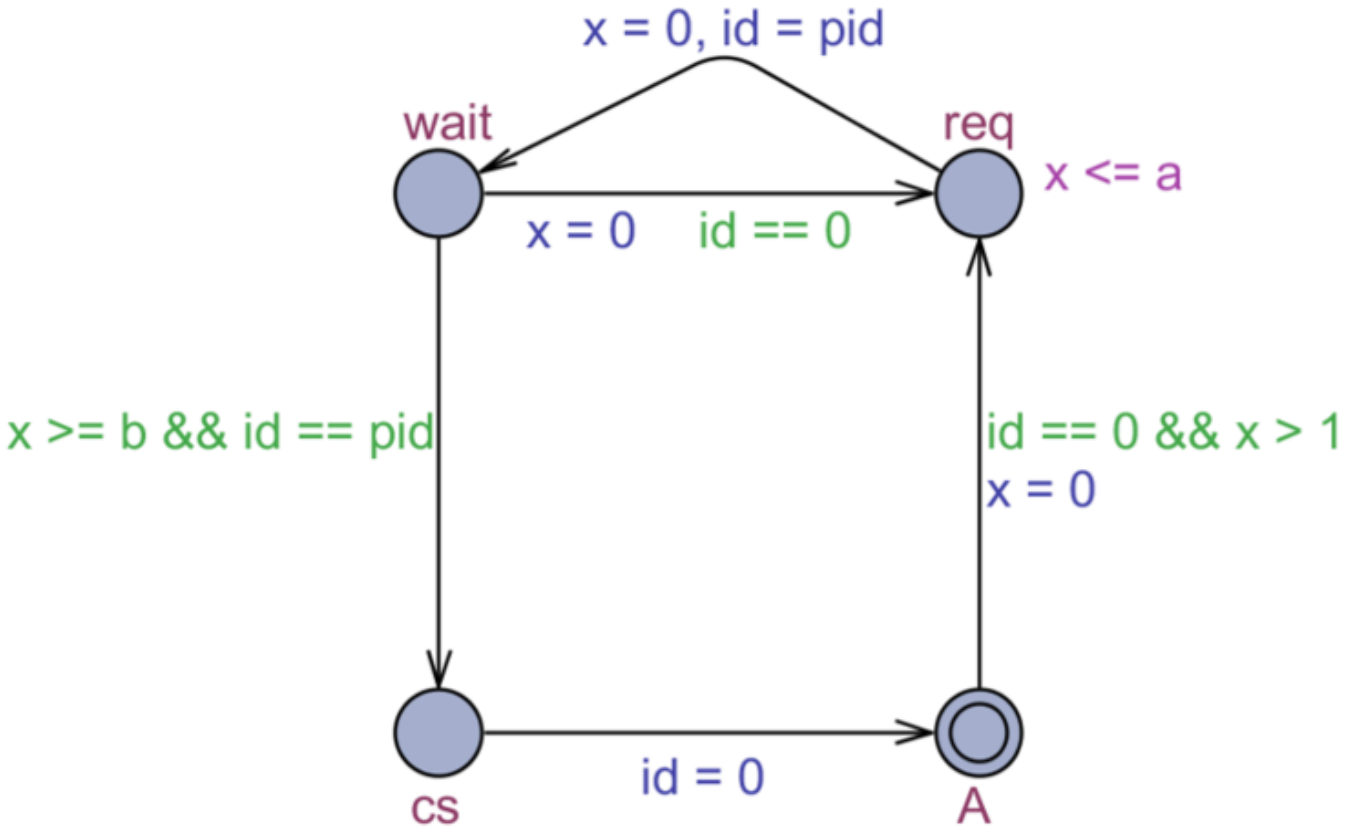
Időzített automaták hálózata

- Időzített automata: véges automata + óraváltozók
  - Állapot: aktív vezérlési hely + óraváltozók állása
- Automaták hálózata
  - Automaták szinkronizációja



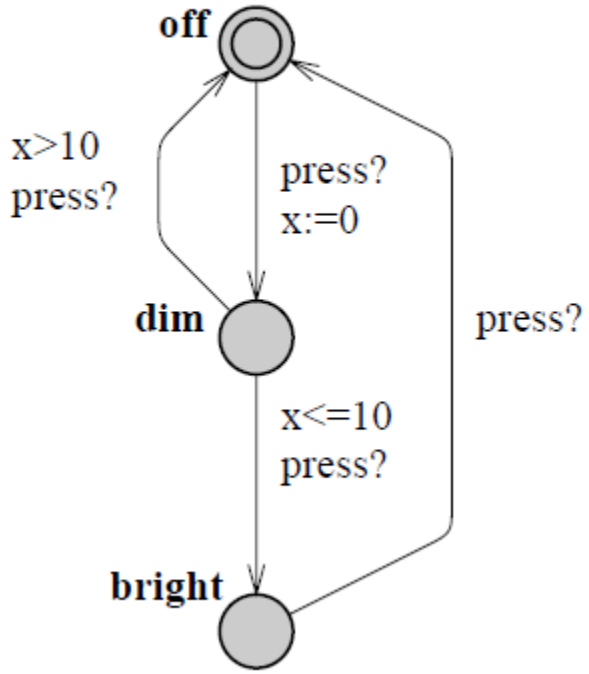
Időzített automaták hálózata

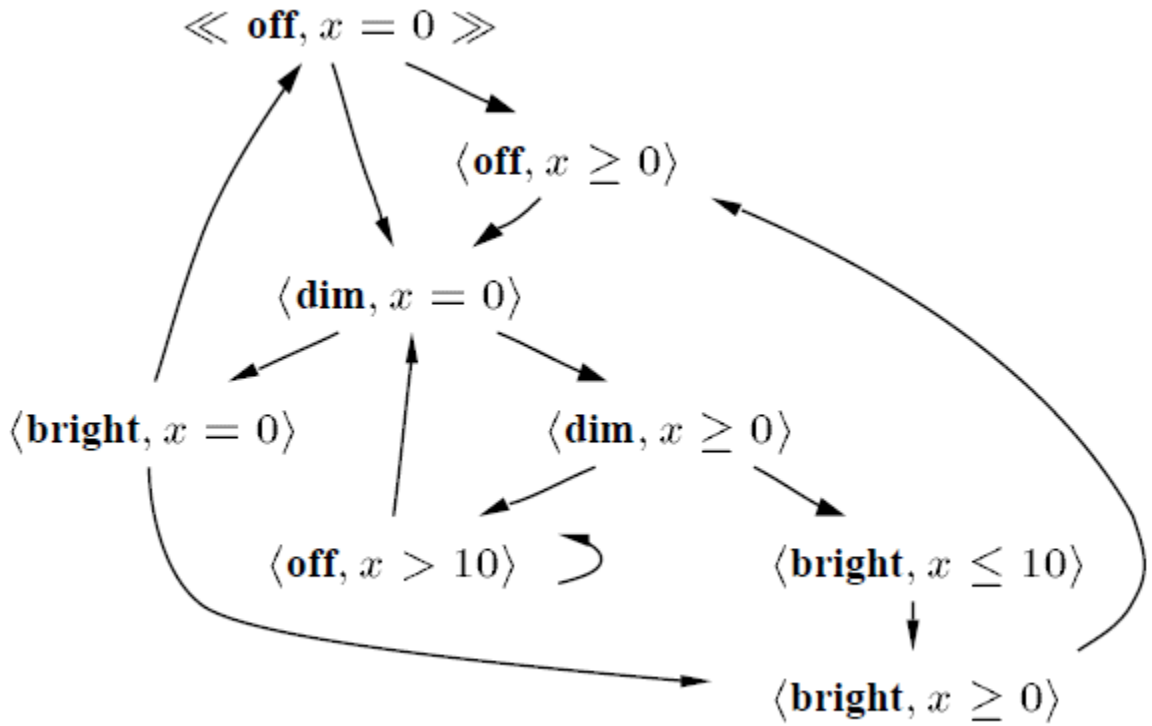
- Időzített automata: véges automata + óraváltozók
  - Állapot: aktív vezérlési hely + óraváltozók állása
- Automaták hálózata
  - Automaták szinkronizációja
- Óraváltozók használata
  - Tranzíció őrfeltétel
  - Lenullázás tranzícióval
  - Vezérlési hely invariáns



Absztrakt reprezentáció

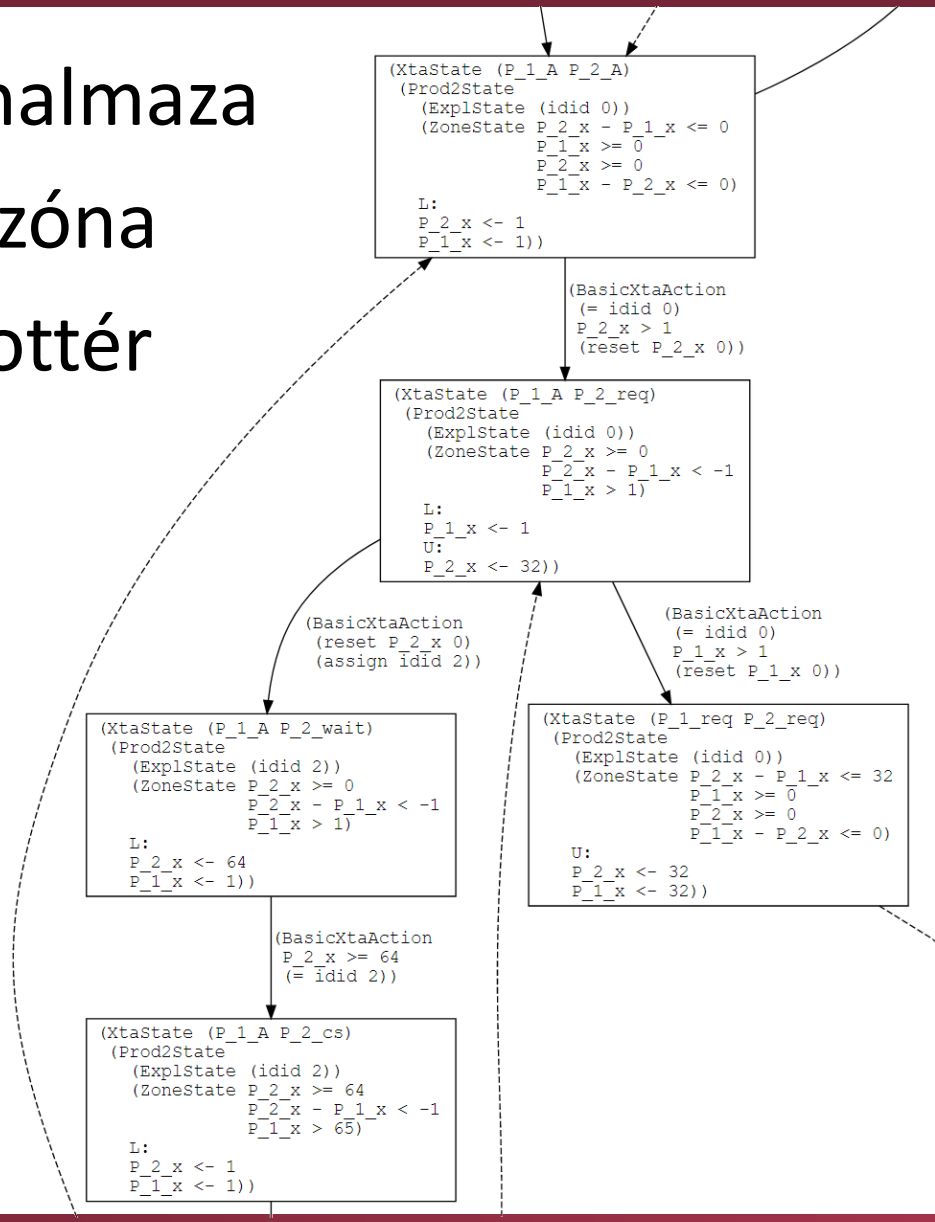
- Zóna: óraváltozókra vonatkozó kényszerek halmaza
- Absztrakt állapot: aktív vezérlési hely(ek) + zóna





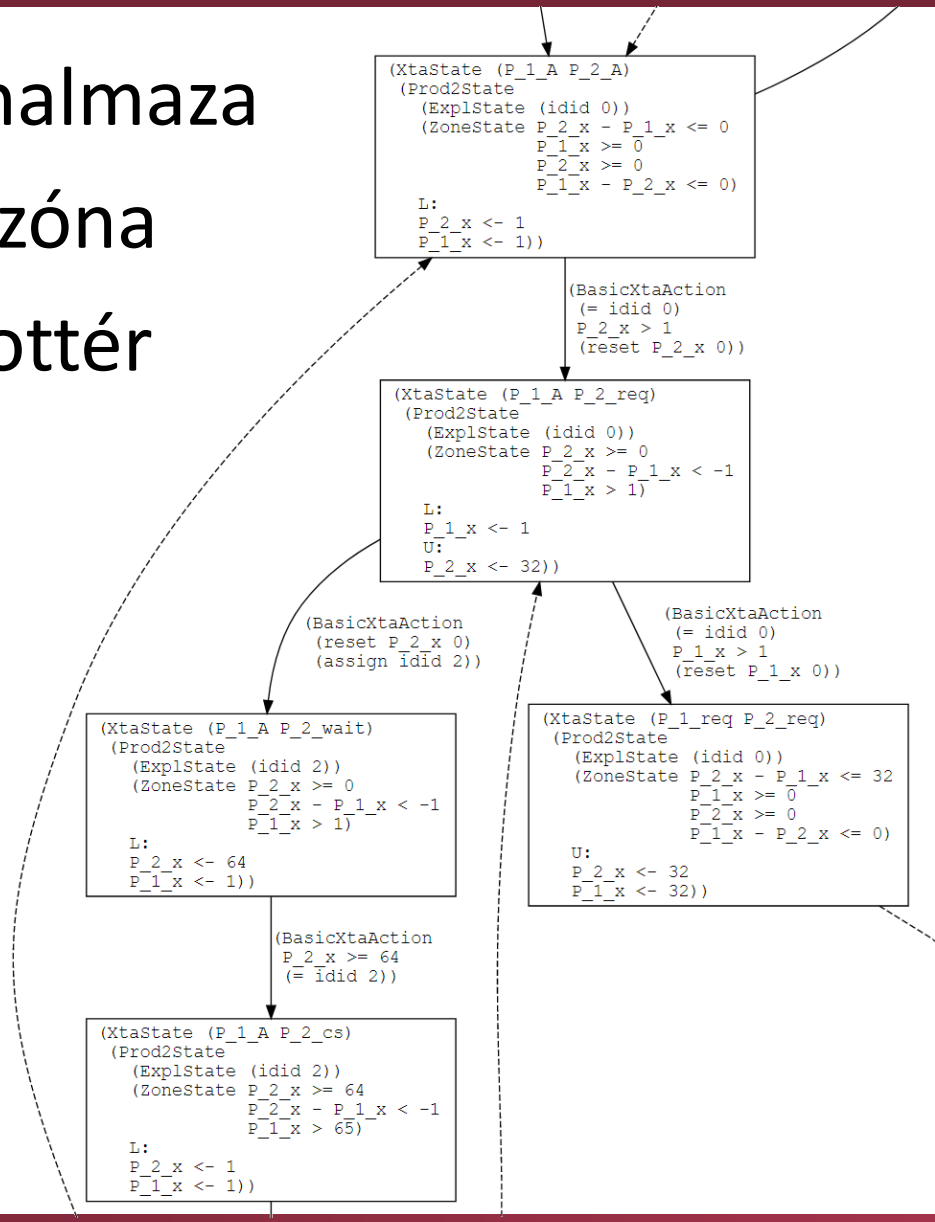
Absztrakt reprezentáció

- Zóna: óraváltozókra vonatkozó kényszerek halmaza
- Absztrakt állapot: aktív vezérlési hely(ek) + zóna
- Adaptive Simulation Graph: absztrakt állapottér
  - Irányított fa gráf
  - Csúcsok: absztrakt állapotok
  - Élek: absztraktállapot-átmenetek
  - *Fedési reláció*



Absztrakt reprezentáció

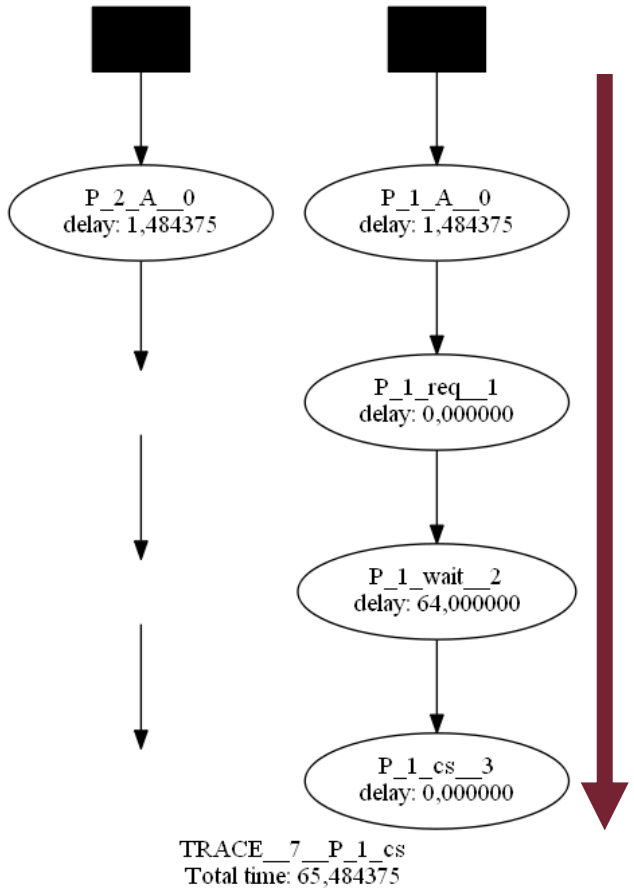
- Zóna: óraváltozókra vonatkozó kényszerek halmaza
- Absztrakt állapot: aktív vezérlési hely(ek) + zóna
- Adaptive Simulation Graph: absztrakt állapottér
  - Irányított fa gráf
  - Csúcsok: absztrakt állapotok
  - Élek: absztraktállapot-átmenetek
  - *Fedési reláció*
- ASG-ből kiolvasható az állapotok (vezérlési helyek) elérhetősége

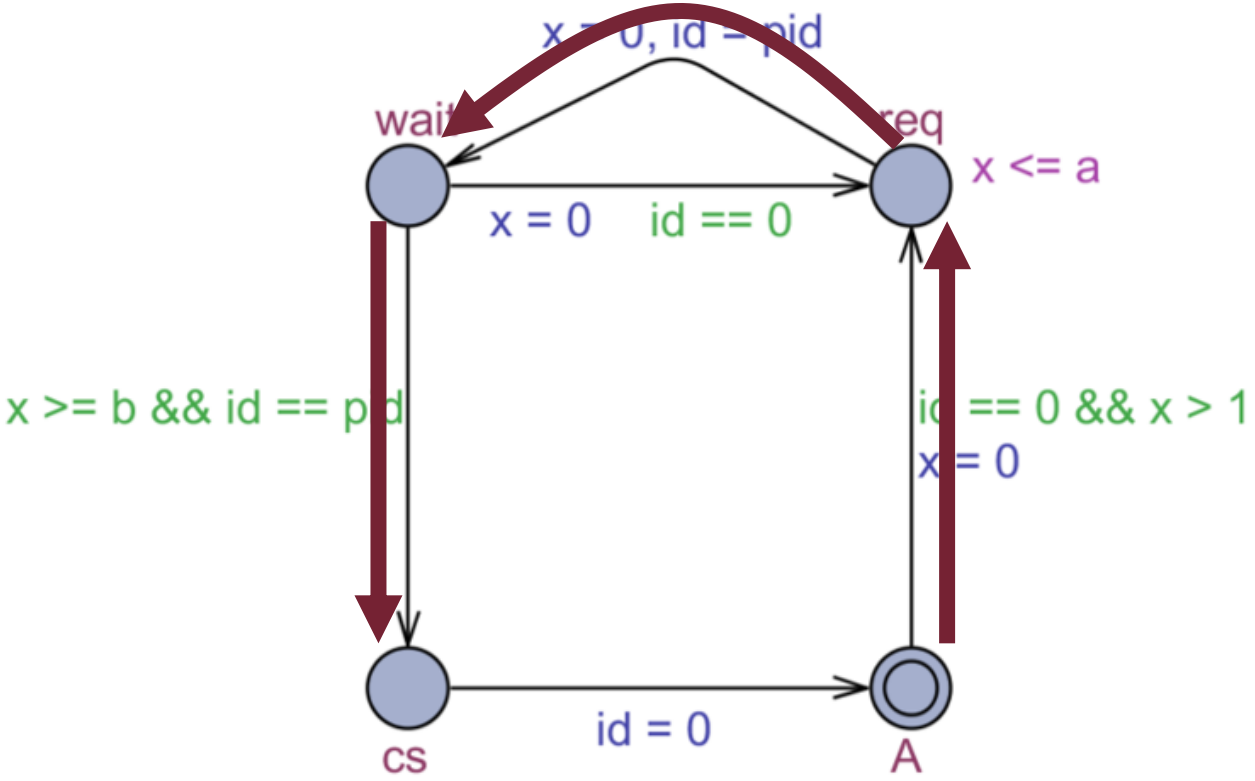


A tesztkészlet elvárt tulajdonságai

A tesztkészlet elvárt tulajdonságai

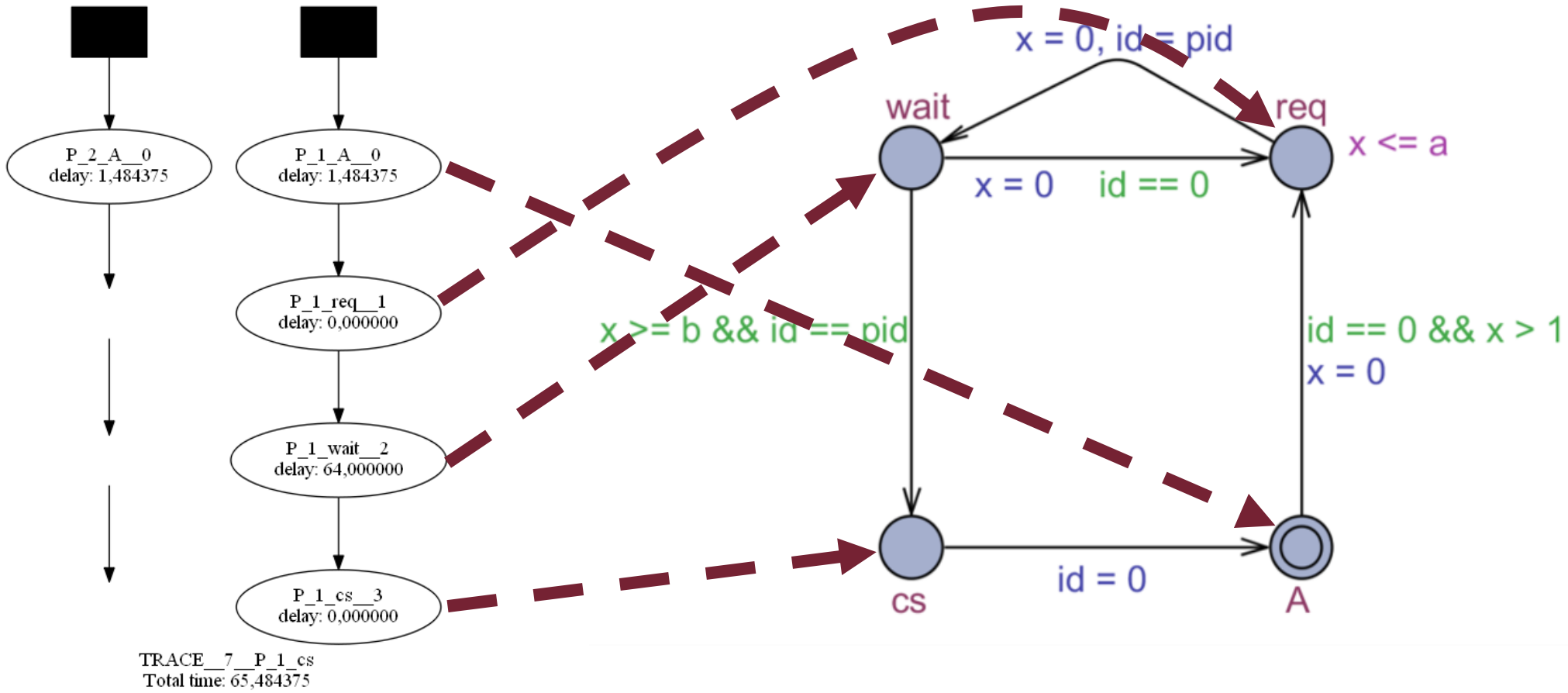
1. A tesztesetek legyenek valódiak





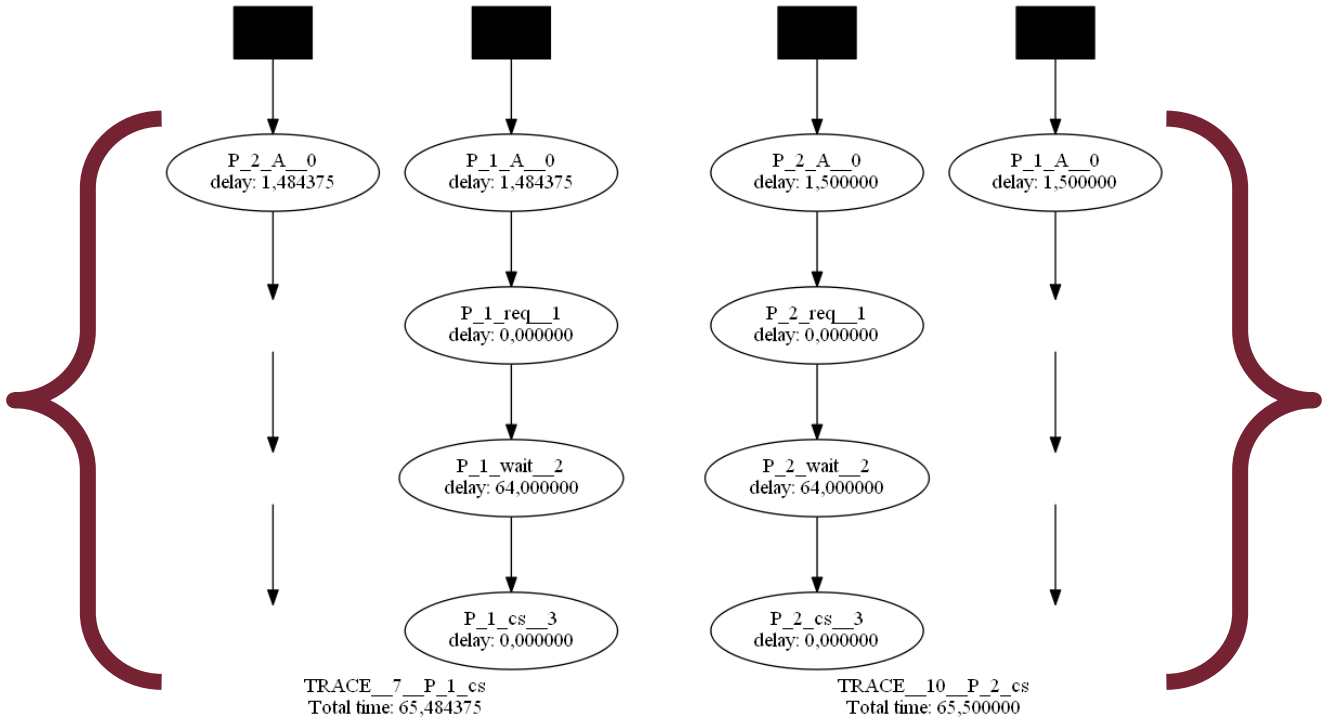
A tesztkészlet elvárt tulajdonságai

1. A tesztesetek legyenek valódiak
2. A tesztkészlet fedje le az összes **elérhető** vezérlési helyet



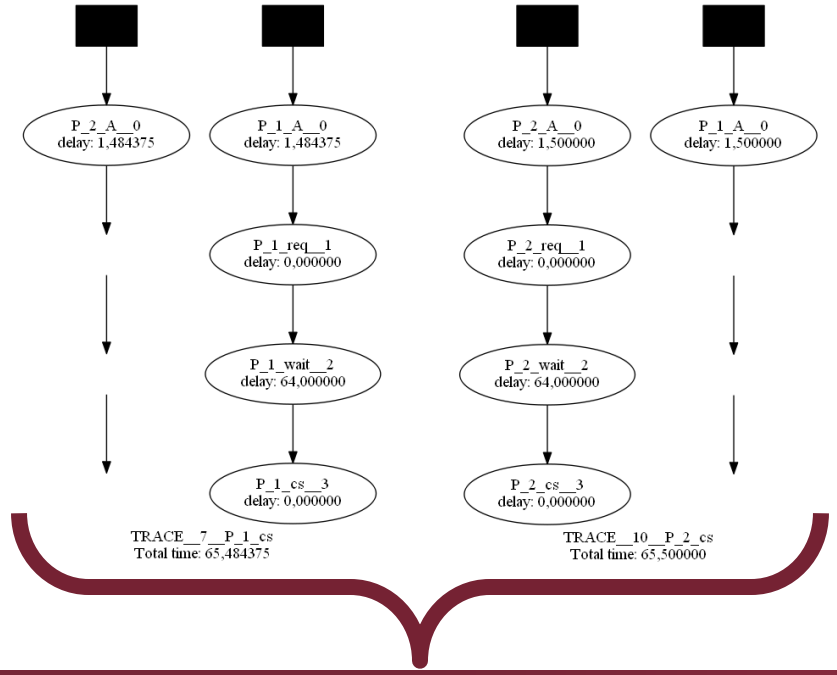
A tesztkészlet elvárt tulajdonságai

1. A tesztesetek legyenek valódiak
2. A tesztkészlet fedje le az összes **elérhető** vezérlési helyet
3. A tesztesetek minél kevesebb lépésből álljanak



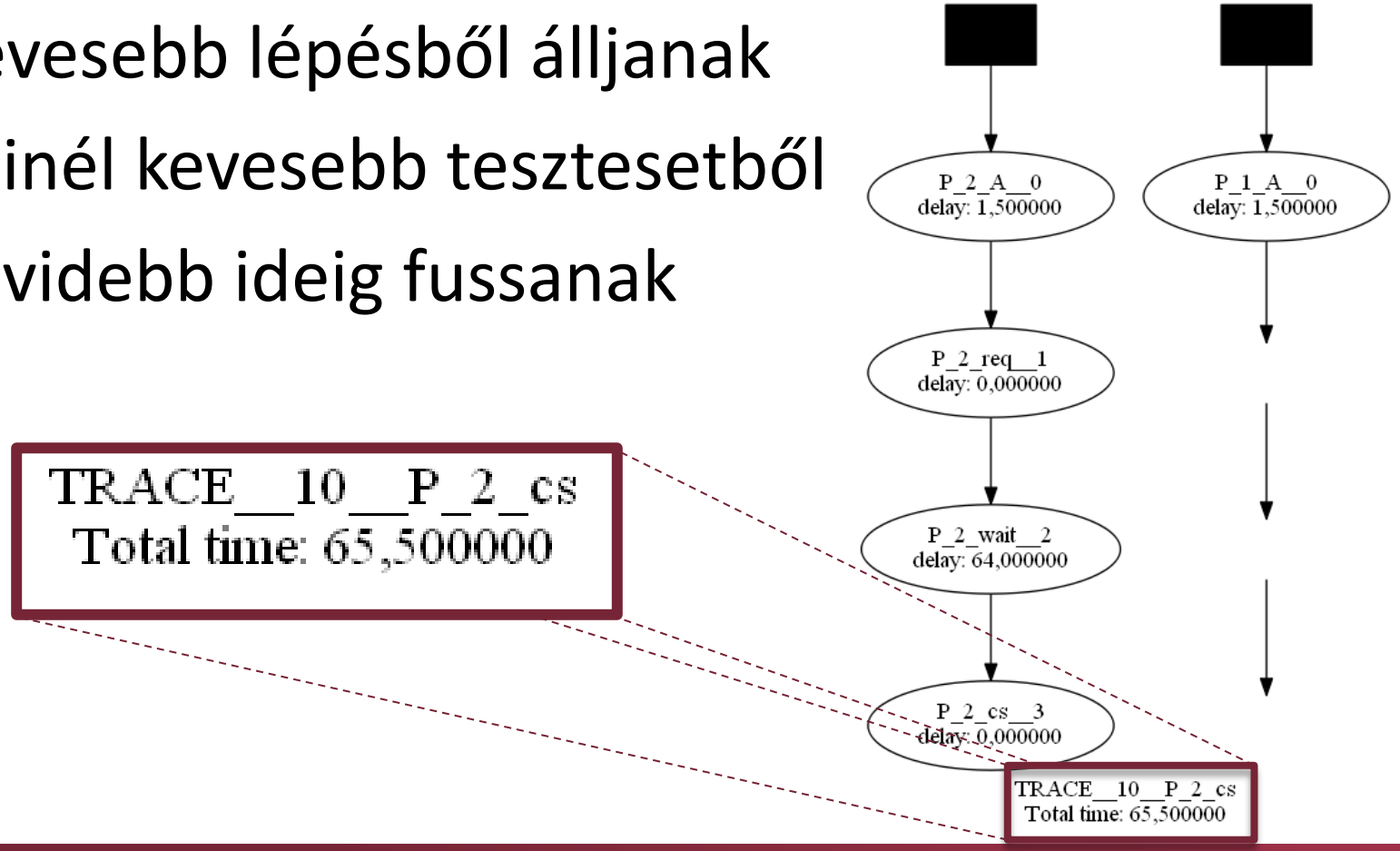
A tesztkészlet elvárt tulajdonságai

1. A tesztesetek legyenek valódiak
2. A tesztkészlet fedje le az összes **elérhető** vezérlési helyet
3. A tesztesetek minél kevesebb lépésből álljanak
4. A tesztkészlet álljon minél kevesebb tesztesetből



A tesztkészlet elvárt tulajdonságai

1. A tesztesetek legyenek valódiak
2. A tesztkészlet fedje le az összes **elérhető** vezérlési helyet
3. A tesztesetek minél kevesebb lépésből álljanak
4. A tesztkészlet álljon minél kevesebb tesztesetből
5. A tesztesetek minél rövidebb ideig fussanak



A tesztek útvonala

- Absztrakt teszteset: gyökérből induló út az ASG-ben
  - ASG-csúcsok és ASG-élek alternáló sorozata
  - Vezérlési helyek (vezérlésihely-vektorok) és tranzíciók (tranzícióvektorok) alternáló sorozata

A tesztek útvonala

- Absztrakt teszteset: gyökérből induló út az ASG-ben
  - ASG-csúcsok és ASG-élek alternáló sorozata
  - Vezérlési helyek (vezérlésihely-vektorok) és tranzíciók (tranzícióvektorok) alternáló sorozata
- ASG bejárása
  - Szélességi bejárás
  - Amíg minden elérhető vezérlési helyet le nem fedünk tesztesettel

A tesztek időzítése

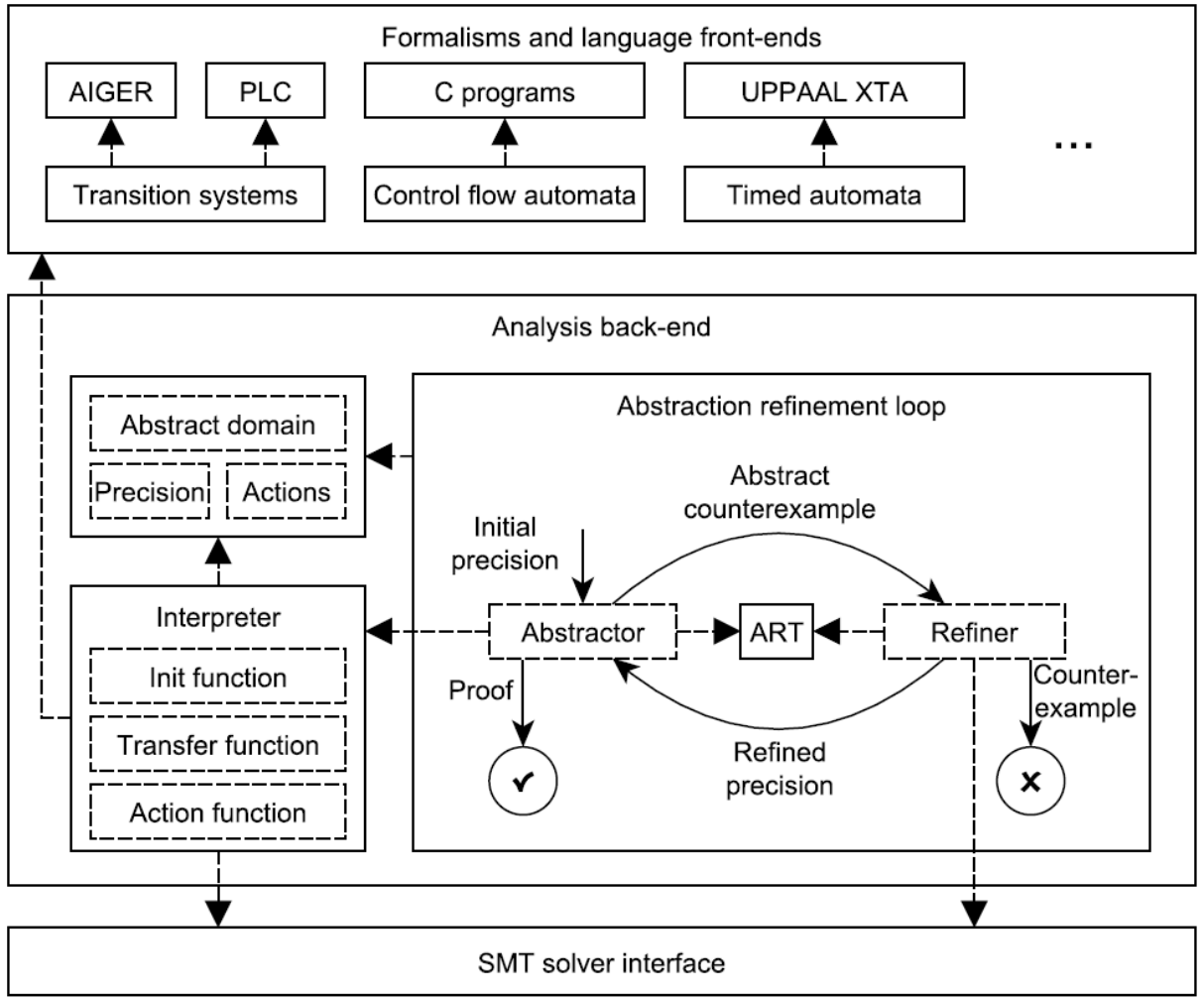
- Valós idejű teszteset: absztrakt teszteset + időzítés
- Időzítés: az állapotokban (tesztlépésekben) eltöltött idők sorozata

A tesztek időzítése

- Valós idejű teszteset: absztrakt teszteset + időzítés
- Időzítés: az állapotokban (tesztlépésekben) eltöltött idők sorozata
- Egyenlőtlenségrendszer megoldása (SMT probléma)
  - Változók: óraváltozók (pl. x) a teszt minden lépésében (pl. $x_1, \ldots, x_n$)
  - Egyenlőtlenségek
    - Absztrakt állapotok zónái
      - Invariánsok
      - Őrfeltételek
    - Kapcsolat az óraváltozók egymást követő értékei között (pl. x_i és x_{i+1})
      - Tranzíció lenullázza
      - Tranzíció nem nullázza le

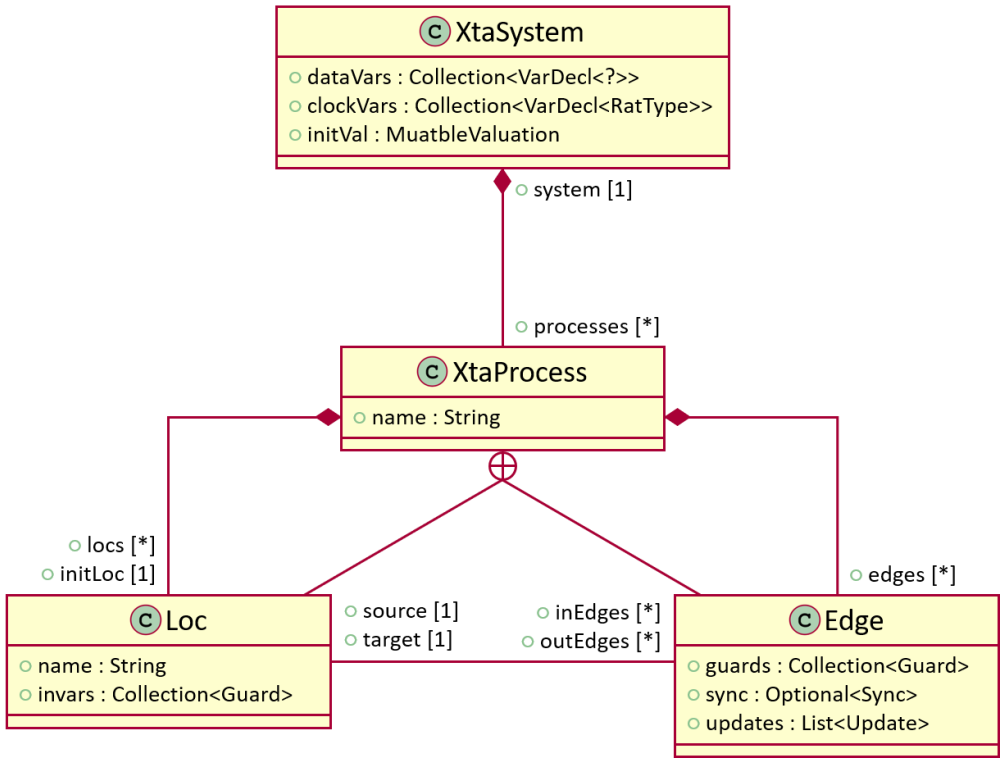
A tesztgenerálás megvalósítása

- A Theta meglévő, felhasznált komponensei



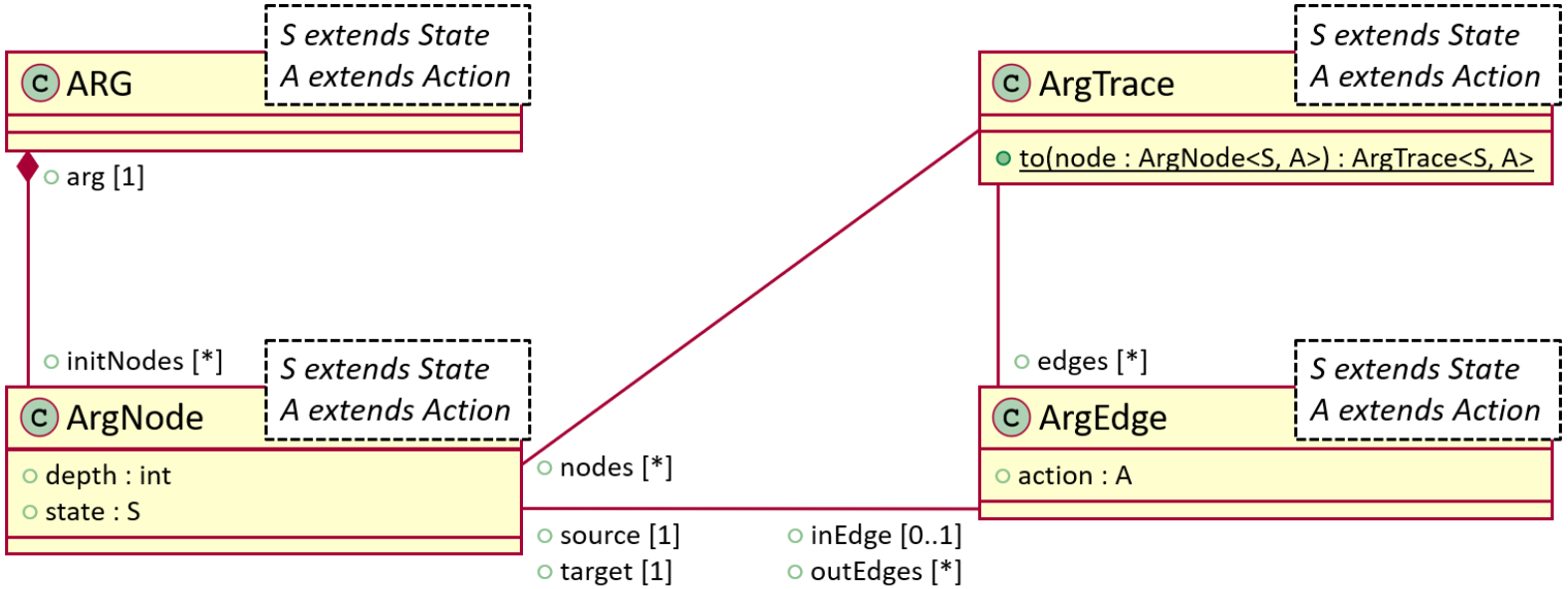
A tesztgenerálás megvalósítása

- A Theta meglévő, felhasznált komponensei
  - XTA formalizmusból előállítja az automatahálózat belső reprezentációját
    - `XtaSystem`, `XtaProcess`, `Loc`, `Edge`



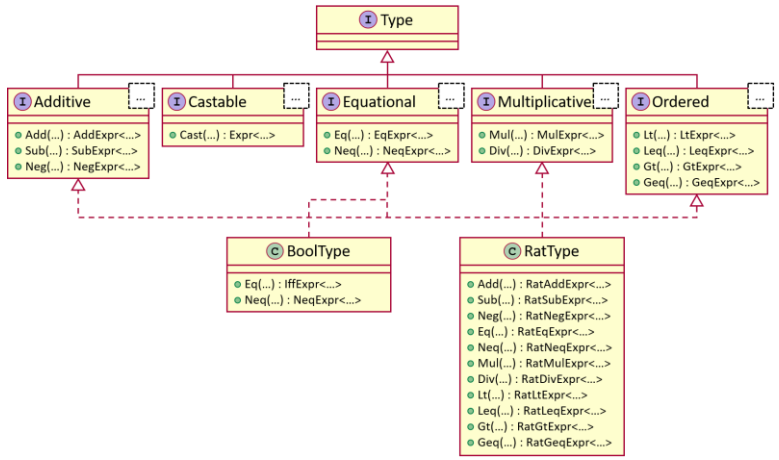
A tesztgenerálás megvalósítása

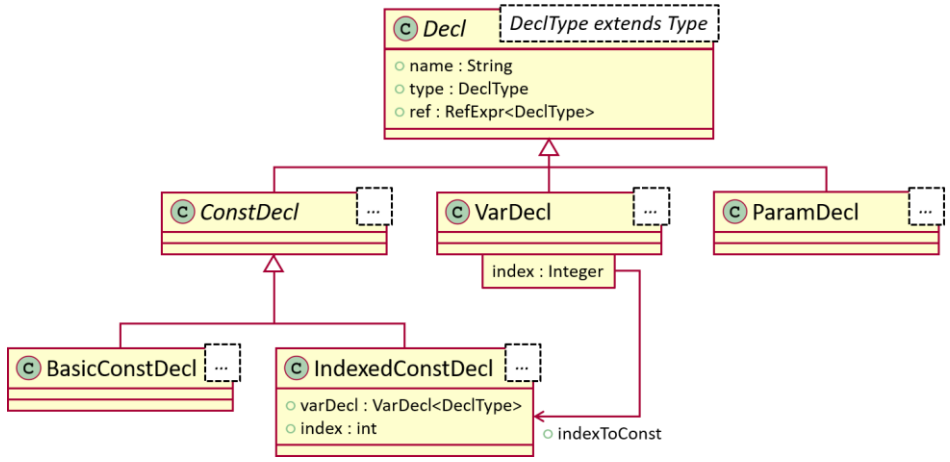
- A Theta meglévő, felhasznált komponensei
  - XTA formalizmusból előállítja az automatahálózat belső reprezentációját
    - `XtaSystem`, `XtaProcess`, `Loc`, `Edge`
  - Előállítja az automatahálózat absztrakt reprezentációját
    - `ARG`, `ArgNode`, `ArgEdge`, `ArgTrace` (`XtaState`, `XtaAction`)

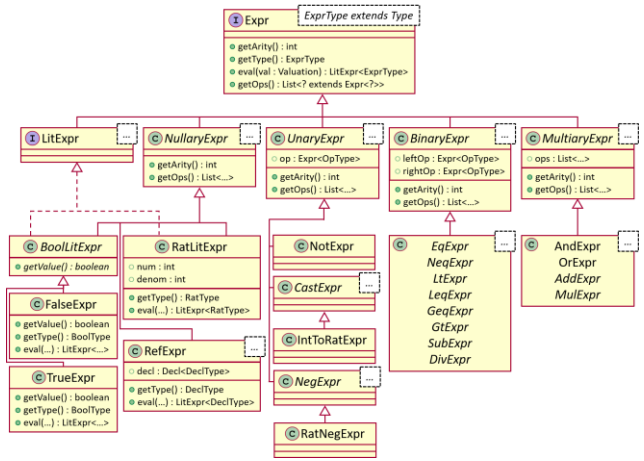


A tesztgenerálás megvalósítása

- A Theta meglévő, felhasznált komponensei
  - XTA formalizmusból előállítja az automatahálózat belső reprezentációját
    - `XtaSystem`, `XtaProcess`, `Loc`, `Edge`
  - Előállítja az automatahálózat absztrakt reprezentációját
    - `ARG`, `ArgNode`, `ArgEdge`, `ArgTrace` (`XtaState`, `XtaAction`)
  - Általános osztályok
    - Típusok, változók, kifejezések







A tesztgenerálás megvalósítása

- A Theta meglévő, felhasznált komponensei
  - XTA formalizmusból előállítja az automatahálózat belső reprezentációját
    - `XtaSystem`, `XtaProcess`, `Loc`, `Edge`
  - Előállítja az automatahálózat absztrakt reprezentációját
    - `ARG`, `ArgNode`, `ArgEdge`, `ArgTrace` (`XtaState`, `XtaAction`)
  - Általános osztályok
    - Típusok, változók, kifejezések
  - SMT solver interfész (Microsoft Z3)

A tesztgenerálás megvalósítása

- A Theta meglévő, felhasznált komponensei
  - XTA formalizmusból előállítja az automatahálózat belső reprezentációját
    - `XtaSystem`, `XtaProcess`, `Loc`, `Edge`
  - Előállítja az automatahálózat absztrakt reprezentációját
    - `ARG`, `ArgNode`, `ArgEdge`, `ArgTrace` (`XtaState`, `XtaAction`)
  - Általános osztályok
    - Típusok, változók, kifejezések
  - SMT solver interfész (Microsoft Z3)
  - Segédosztályok
    - Vizualizáció, logger

A tesztgenerálás megvalósítása

- Saját, új osztályok

A tesztgenerálás megvalósítása

- Saját, új osztályok
  - `XtaTest`: időzített teszteset

A tesztgenerálás megvalósítása

- Saját, új osztályok
  - `XtaTest`: időzített teszteset
  - `XtaTestGenerator`: tesztgenerálási algoritmus implementációja

A tesztgenerálás megvalósítása

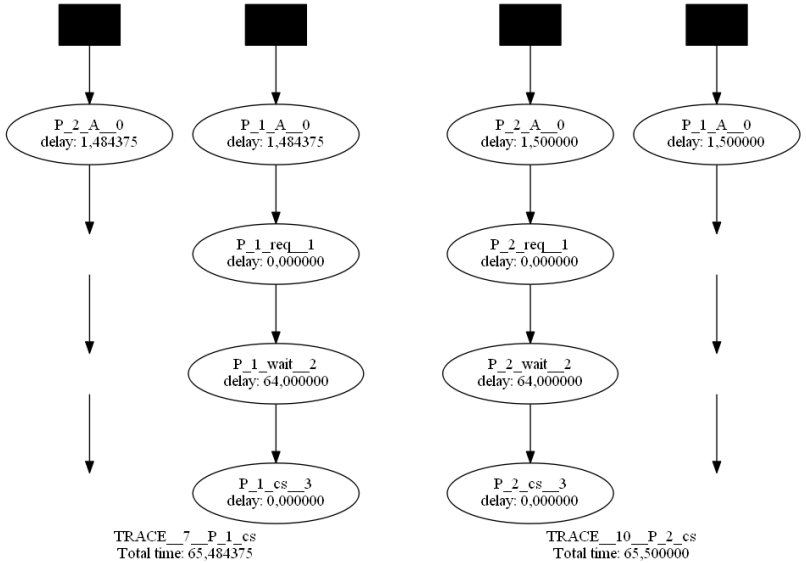
- Saját, új osztályok
  - `XtaTest`: időzített teszteset
  - `XtaTestGenerator`: tesztgenerálási algoritmus implementációja
  - `XtaTestPrinter`: teszteset(ek) szöveges megjelenítése

```
========== TRACE__7__P_1_cs ==========
(XtaState (P_1_A P_2_A)
  (Prod2State
    (ExplState (idid 0))
    (ZoneState P_2_x - P_1_x <= 0
               P_1_x >= 0
               P_2_x >= 0
               P_1_x - P_2_x <= 0)
    L:
    P_2_x <- 1
    P_1_x <- 1))
Delay: 1,484375
(BasicXtaAction
  (= idid 0)
  P_1_x > 1
  (reset P_1_x 0))
(XtaState (P_1_req P_2_A)
  (Prod2State
    (ExplState (idid 0))
    (ZoneState P_1_x - P_2_x < -1
               P_2_x > 1
               P_1_x >= 0)
    L:
    P_2_x <- 1
    U:
    P_1_x <- 32))
Delay: 0,000000
(BasicXtaAction
  (reset P_1_x 0)
  (assign idid 1))
(XtaState (P_1_wait P_2_A)
  (Prod2State
    (ExplState (idid 1))
    (ZoneState P_1_x - P_2_x < -1
               P_2_x > 1
               P_1_x >= 0)
    L:
    P_2_x <- 1
    P_1_x <- 64))
Delay: 64,000000
(BasicXtaAction
  (= idid 1)
  P_1_x >= 64)
(XtaState (P_1_cs P_2_A)
  (Prod2State
    (ExplState (idid 1))
    (ZoneState P_1_x - P_2_x < -1
               P_2_x > 65
               P_1_x >= 64)
    L:
    P_2_x <- 1
    P_1_x <- 1))
Delay: 0,000000
Total time: 65,484375
```

```
========== TRACE__10__P_2_cs ==========
(XtaState (P_1_A P_2_A)
  (Prod2State
    (ExplState (idid 0))
    (ZoneState P_2_x - P_1_x <= 0
               P_1_x >= 0
               P_2_x >= 0
               P_1_x - P_2_x <= 0)
    L:
    P_2_x <- 1
    P_1_x <- 1))
Delay: 1,500000
(BasicXtaAction
  (= idid 0)
  P_2_x > 1
  (reset P_2_x 0))
(XtaState (P_1_A P_2_req)
  (Prod2State
    (ExplState (idid 0))
    (ZoneState P_2_x >= 0
               P_2_x - P_1_x < -1
               P_1_x > 1)
    L:
    P_1_x <- 1
    U:
    P_2_x <- 32))
Delay: 0,000000
(BasicXtaAction
  (reset P_2_x 0)
  (assign idid 2))
(XtaState (P_1_A P_2_wait)
  (Prod2State
    (ExplState (idid 2))
    (ZoneState P_2_x >= 0
               P_2_x - P_1_x < -1
               P_1_x > 1)
    L:
    P_2_x <- 64
    P_1_x <- 1))
Delay: 64,000000
(BasicXtaAction
  P_2_x >= 64
  (= idid 2))
(XtaState (P_1_A P_2_cs)
  (Prod2State
    (ExplState (idid 2))
    (ZoneState P_2_x >= 64
               P_2_x - P_1_x < -1
               P_1_x > 65)
    L:
    P_2_x <- 1
    P_1_x <- 1))
Delay: 0,000000
Total time: 65,500000
```

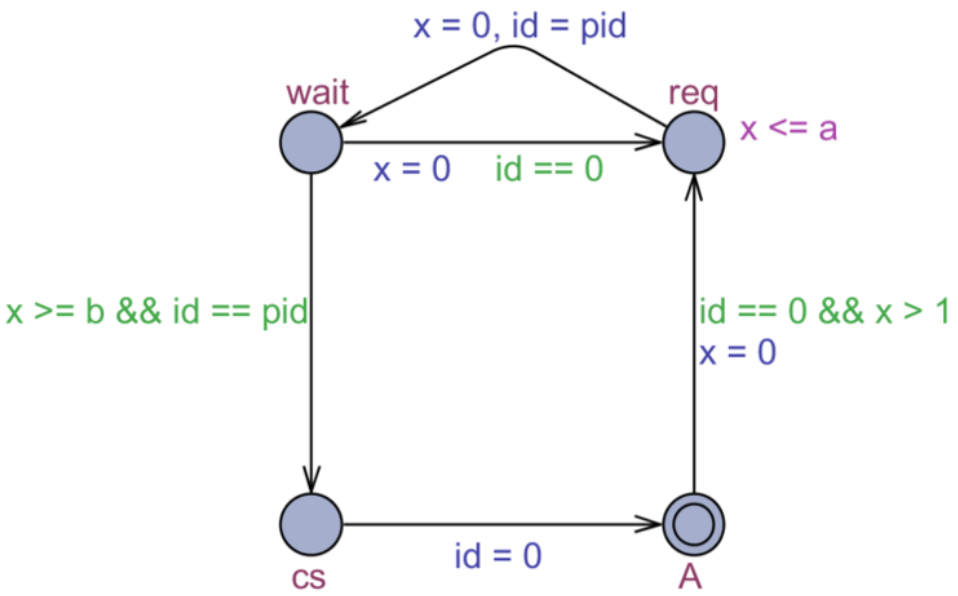
A tesztgenerálás megvalósítása

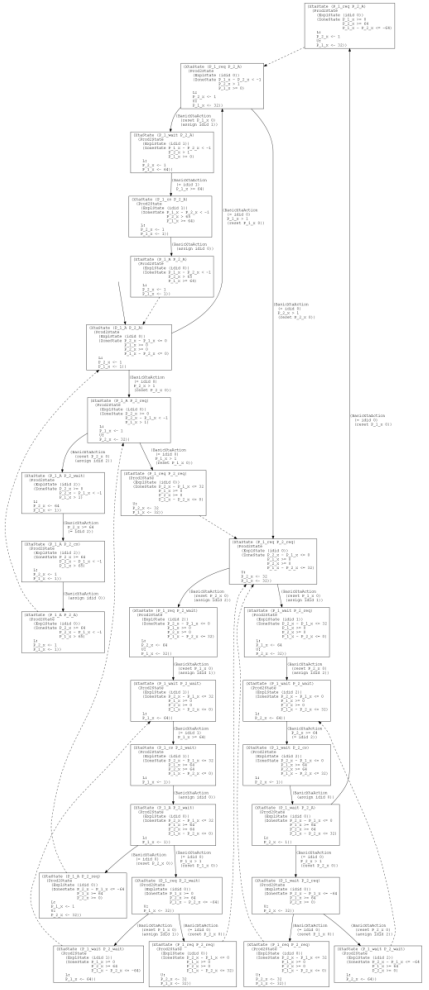
- Saját, új osztályok
  - `XtaTest`: időzített teszteset
  - `XtaTestGenerator`: tesztgenerálási algoritmus implementációja
  - `XtaTestPrinter`: teszteset(ek) szöveges megjelenítése
  - `XtaTestVisualizer`: teszteset(ek) grafikus megjelenítése (Graphviz)

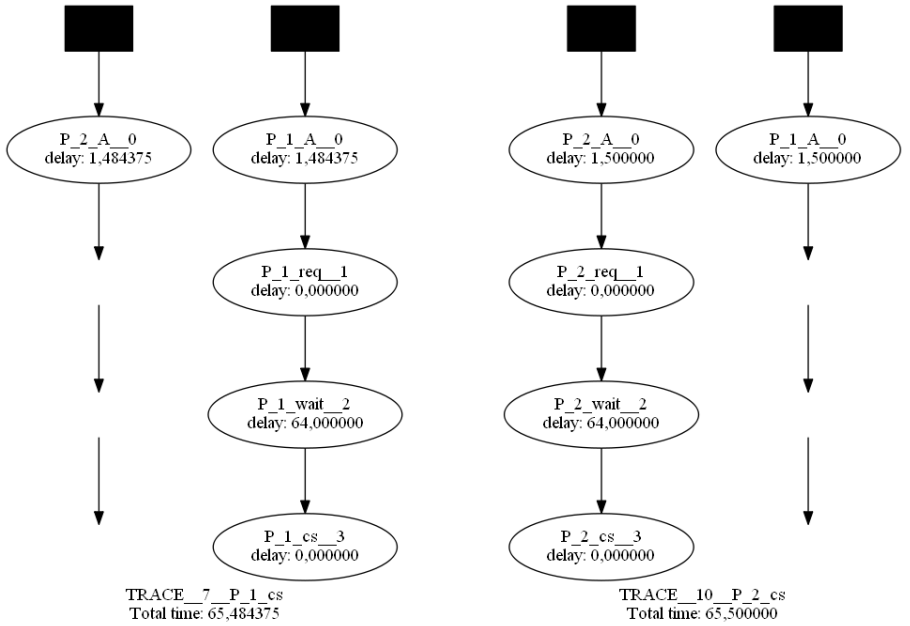


Kiértékelés

- Esettanulmány: Fischer-protokoll







Kiértékelés

- Esettanulmány: Fischer-protokoll
- Mérések

Modell	ARG mélység	ARG méret	Futásidő (ms)	Generált tesztek	$\lvert\hat{\mathfrak{T}}\rvert$	$\sum \lvert\hat{\mathfrak{T}}_i\rvert$
critical-01-25-50.xta	7	27	274	12	2	13
critical-02-25-50.xta	14	641	1723	85	4	28
critical-03-25-50.xta	22	21699	7623	410	6	45
csma-01.xta	2	3	509	2	1	2
csma-02.xta	5	21	353	9	2	7
csma-03.xta	8	99	1306	18	4	13
csma-04.xta	9	381	875	24	6	19
csma-05.xta	10	1272	1723	40	7	21
csma-06.xta	11	3865	2040	53	9	25
csma-07.xta	12	11008	4488	93	12	33
csma-08.xta	13	29925	4264	100	13	37
csma-09.xta	14	78552	4278	119	16	44
fddi-001.xta	9	15	813	11	2	15
fddi-002.xta	16	46	2144	26	3	34
fddi-003.xta	24	104	8495	48	4	60
fddi-004.xta	28	101	18073	62	4	74
fddi-005.xta	34	141	52396	85	4	89
fddi-006.xta	időtúllépés					
fischer-01-32-64.xta	4	5	152	4	1	4
fischer-02-32-64.xta	8	27	380	10	2	8
fischer-03-32-64.xta	10	121	508	20	3	12
fischer-04-32-64.xta	12	493	1083	35	4	16
fischer-05-32-64.xta	14	1911	1866	56	5	20
fischer-06-32-64.xta	16	7183	2648	79	6	24
fischer-07-32-64.xta	18	26405	3294	104	7	28
fischer-08-32-64.xta	20	95353	5829	165	8	32
lynch-01-16.xta	9	10	296	9	1	9
lynch-02-16.xta	13	61	867	30	2	18
lynch-03-16.xta	15	271	1716	76	3	27
lynch-04-16.xta	17	1049	4395	208	4	36
lynch-05-16.xta	19	3811	13247	582	5	45
lynch-06-16.xta	21	13453	39038	1490	6	54
lynch-07-16.xta	időtúllépés					
lynch-08-16.xta	időtúllépés					

Kiértékelés

- Esettanulmány: Fischer-protokoll
- Mérések
- Továbbfejlesztési lehetőségek
  - Tesztkészlet
    - Tesztkészlettel szemben támasztott elvárások priorizálása
    - Új elvárások a tesztkészlettel szemben (pl. zóna közepe/széle)
  - Kimeneti formátum
    - UPPAAL szimulátor
    - Tesztesetek tényleges, futtatható előállítása

Összefoglalás

- A feladat kontextusba helyezése
- Háttérismeretek bemutatása (időzített automata, zóna, ASG)
- A tesztkészlet elvárt tulajdonságai
- A tesztek útvonala
- A tesztek időzítése
- A tesztgenerálás megvalósítása
- Kiértékelés

BÍRÁLÓI KÉRDÉSEK

1. kérdés

Tud említeni olyan formális módszert, melynek sikeres alkalmazásakor a vizsgált rendszer tesztelése biztonságosan kihagyható?

- Önmagában a modellek helyességének bizonyítása nem elegendő, hiszen az implementációba is csúszhatnak hibák
- A tesztelés kihagyása csak akkor jöhet szóba, ha az implementáció (kódgenerálás) is teljes mértékben automatikusan (és bizonyítottan helyesen) történik

2. kérdés

A kielégíthetőségi probléma, melyre a teszteset-konkretizálás a dolgozat 3.5 fejezetében visszavezetésre kerül, melyik elméletben értelmezett?

- Az óraváltozókat valós értékkészlettel definiáltam
- Az implementációmban racionális számokként kezelem őket
- Az algoritmus helyessége ettől független, hiszen egy valós/racionális különbség nem okozhat változást, ha minden óraváltozóra vonatkozó kényszerben egész szám szerepel.

3. kérdés

A `calculateDelays` függvényen belül az SMT-probléma összeállításakor szükséges az `XtaState` példányokat leíró kényszereket hozzávenni a formulahalmazhoz?

- Nem, ez felesleges lépés. Az `XtaState` példányokat leíró kényszerekben csak az szerepel, hogy a zónájuk ne legyen üres. Az XTA modellből származó érdemi kényszerek mind az `XtaAction`-öket leíró kényszerekben találhatók.

4. kérdés

Mennyiben kellene módosítani, illetve bővíteni az elkészült eszközt ahhoz, hogy a 6.2 fejezetben vázolt további fedési kritériumokat is támogassa („Törekedhetünk például olyan tesztesetekre is, amelyek az állapotátmeneteket azok lehetséges időintervallumának a közepén vagy éppen a legszélén tüzelik”)?

- Az időzítést konkretizáló függvényt kell módosítani. A bővítés szükséges mértéke az SMT megoldó képességeitől függ: olyan megoldóra lenne szükség, amely képes szélsőérték-keresésre is.
- A zónák „szélei” (pl. $x_{i_{min}}$ és $x_{i_{max}}$) így meghatározhatók
- Az ezektől vett távolságot kell minimalizálni/maximalizálni