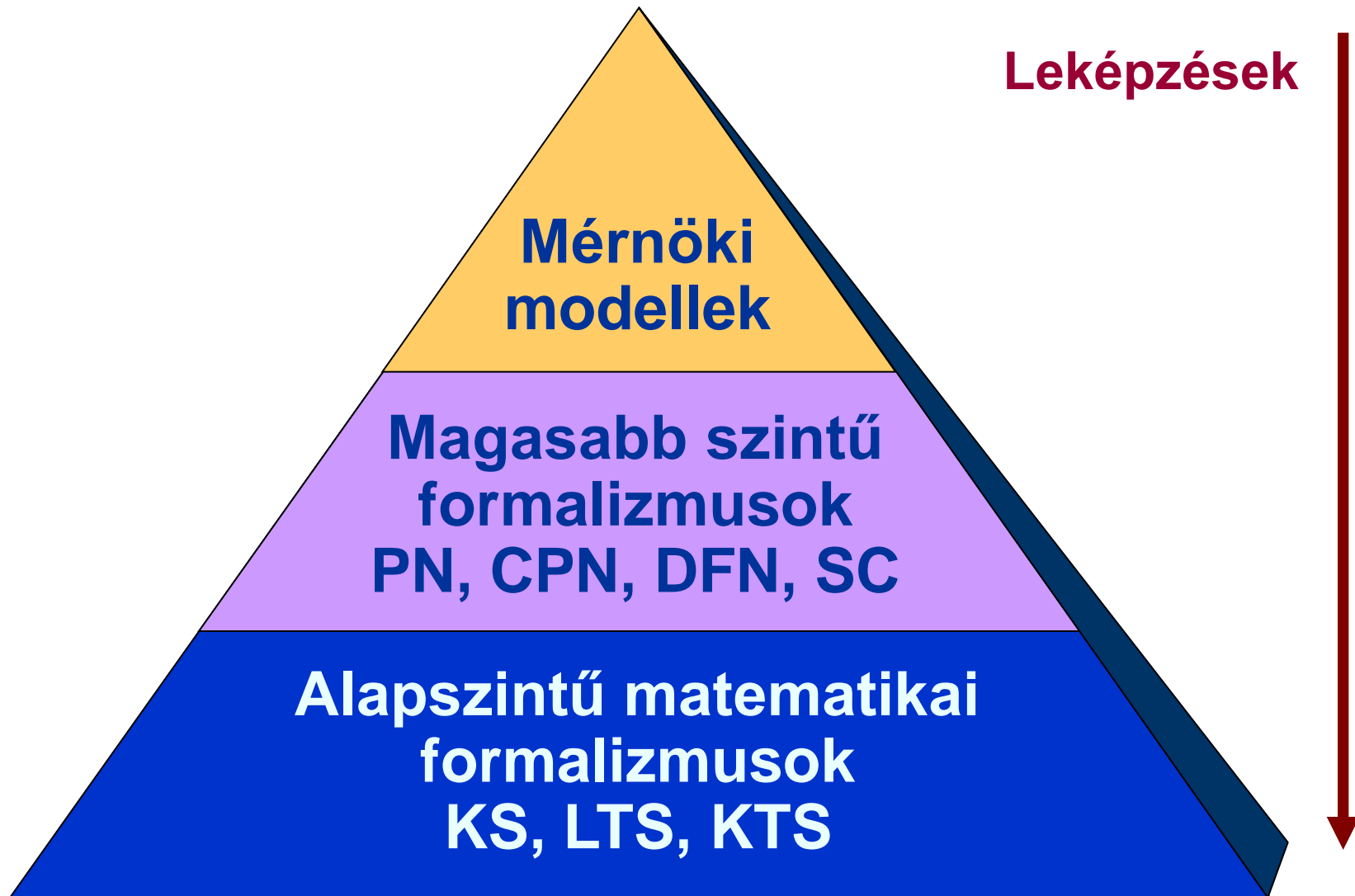


Alapszintű formalizmusok

dr. Majzik István

BME Méréstechnika és Információs Rendszerek Tanszék

Modellek a formális ellenőrzéshez



Alapszintű formalizmusok (áttekintés)

- Kripke-struktúrák (KS)
 - Állapotok, állapotátmenetek
 - Állapotok lokális tulajdonságai mint címkék
- Címkézett tranzíciós rendszerek (LTS)
 - Állapotok, állapotátmenetek
 - Állapotok lokális tulajdonságai mint címkék
- Kripke tranzíciós rendszerek (KTS)
 - Állapotok, állapotátmenetek
 - Állapotok és lokális tulajdonságai mint címkék
- Véges állapotú automaták időkezeléssel
 - Kiterjesztések: Változók, óraváltozók, szinkronizáció

1. Kripke-struktúra

KS, Kripke-structure:

- **Állapotok** tulajdonságait fejezzük ki:
címkézés **atomi kijelentésekkel**
- Egy állapothoz sok címke rendelhető

Alkalmazás: Viselkedés, algoritmus leírása

$KS = (S, R, L)$ és AP , ahol

$AP = \{P, Q, R, \dots\}$ atomi kijelentések halmaza (domén-specifikus)

$S = \{s_1, s_2, s_3, \dots, s_n\}$ állapotok halmaza

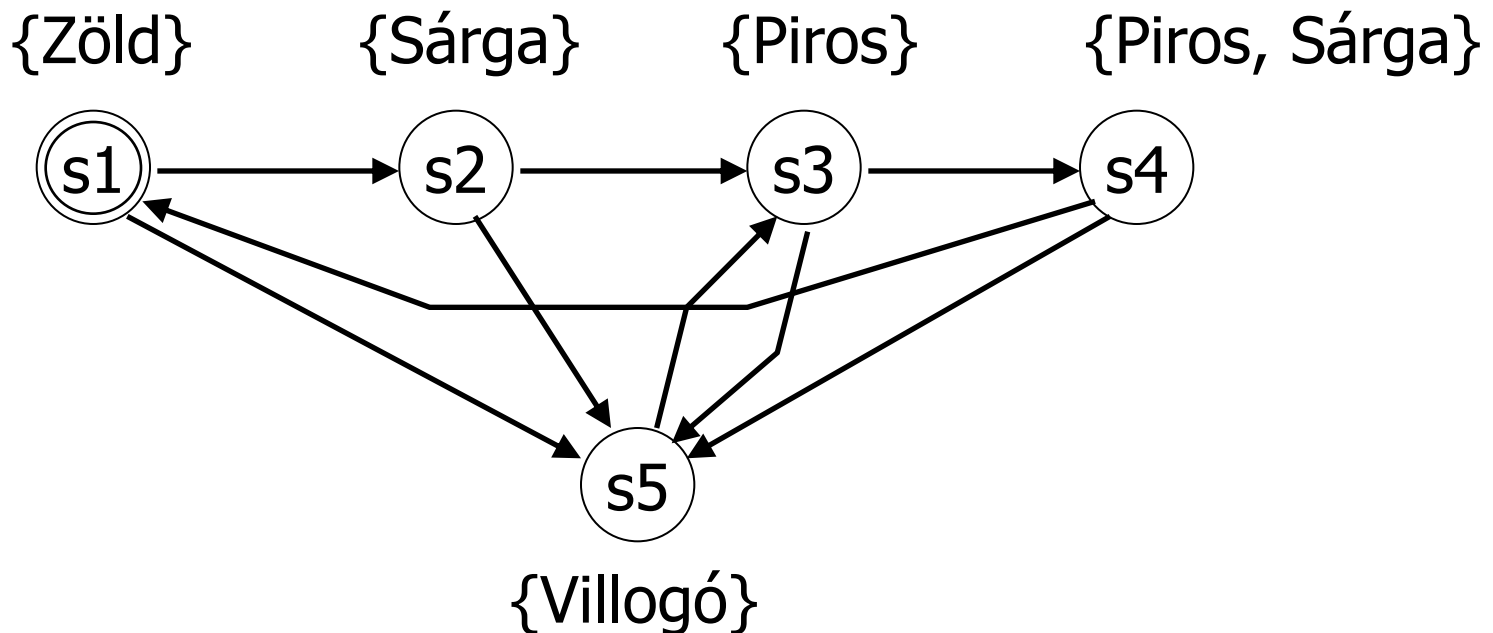
$R \subseteq S \times S$: állapotátmeneti reláció

$L: S \rightarrow 2^{AP}$ állapotok címkézése atomi kijelentésekkel

Kripke-struktúra példa

Közlekedési lámpa viselkedése

- $AP = \{\text{Zöld}, \text{Sárga}, \text{Piros}, \text{Villogó}\}$
- $S = \{s1, s2, s3, s4, s5\}$



2. Címkezett tranzíciós rendszer

LTS, Labeled Transition System:

- **Állapotátmenetek** tulajdonságait fejezzük ki: **címkezés akciókkal**
- Egy átmeneten csak egy akció szerepelhet

Alkalmazás: Kommunikáció, protokollok modellezése

$LTS = (S, Act, \rightarrow)$, ahol

$S = \{s_1, s_2, \dots, s_n\}$ állapotok halmaza

$Act = \{a, b, c, \dots\}$ akciók (címkek) halmaza

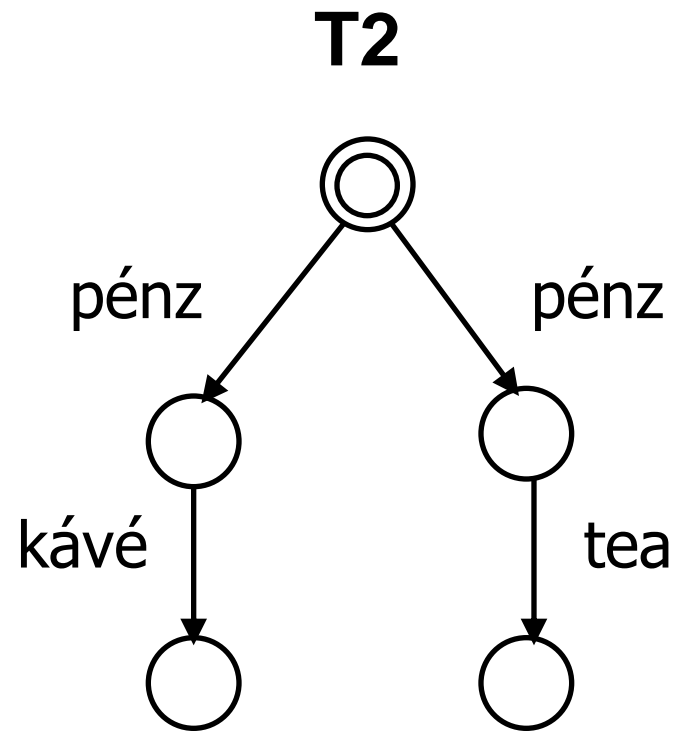
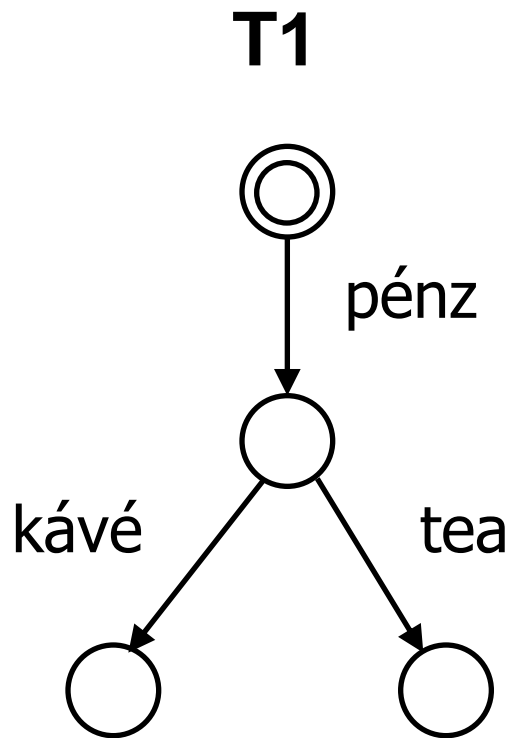
$\rightarrow \subseteq S \times Act \times S$ címkezett állapotátmenetek

Állapotátmenetek szokásos jelölése: $s_1 \xrightarrow{a} s_2$

LTS példák

- Italautomata modelljei

Act = {pénz, kávé, tea}



3. Kripke tranzíciós rendszer

KTS, Kripke Transition System:

- **Állapotok és átmenetek tulajdonságait is kifejezzük: címkézés atomi kijelentésekkel és akciókkal**
- **Egy állapothoz sok címke rendelhető, egy átmenethez egy címke rendelhető**

$KTS = (S, \rightarrow, L)$ és AP, Act , ahol

$AP = \{P, Q, R, \dots\}$ atomi kijelentések halmaza (domén-specifikus)

$Act = \{a, b, c, \dots\}$ akciók halmaza

$S = \{s_1, s_2, s_3, \dots, s_n\}$ állapotok halmaza

$\rightarrow \subseteq S \times Act \times S$ állapotátmeneti reláció

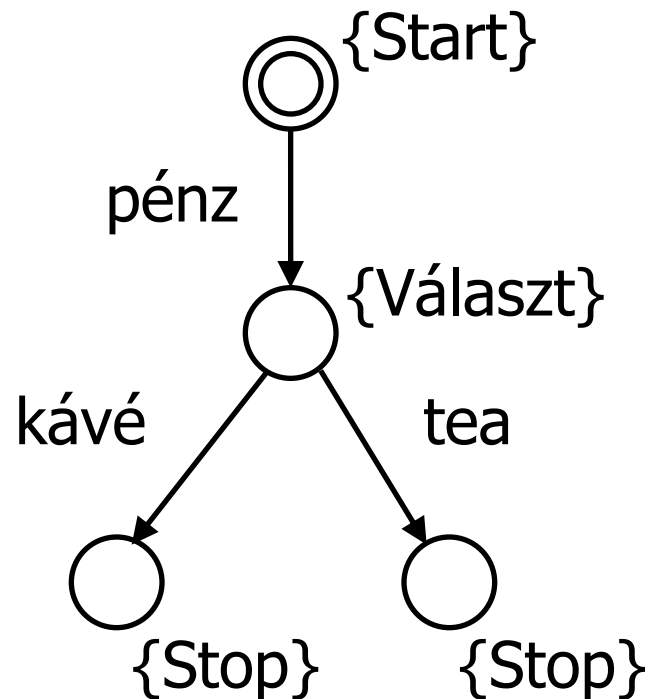
$L: S \rightarrow 2^{AP}$ állapotok címkézése atomi kijelentésekkel

KTS példa

- Italautomata modellje állapot címkékkel

Act = {péNZ, kávé, tea}

AP = {Start, Választ, Stop}



Időzített automaták és az UPPAAL eszköz

Automaták és változók

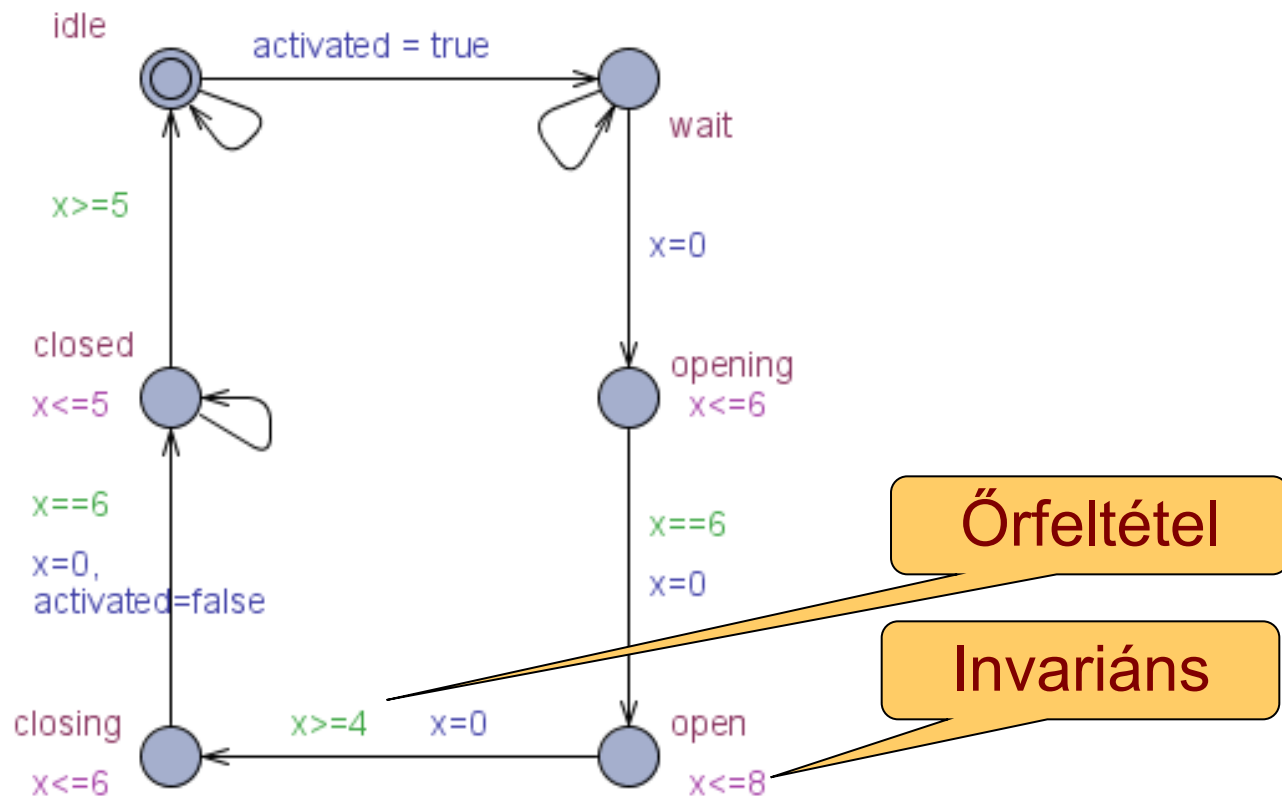
- Cél: Állapot alapú viselkedés modellezése
- Alap formalizmus: Véges állapotú automata (FSM)
 - Állapotok (névvel hivatkozhatók)
 - Állapotátmenetek
- Kiterjesztés: **Egész értékű változók használata**
 - Változók értéktartománya megadható
 - Konstansok definiálhatók
 - Egész aritmetika használható
- Állapotátmenetek kiterjesztése:
 - **Őrfeltétel** hozzárendelése: A változókon kiértékelhető predikátum
 - Az átmenet bekövetkezéséhez igaz kell legyen
 - **Akció** hozzárendelése: Értékadás változóknak

Kiterjesztések óraváltozókkal

- Cél: Valós idejű viselkedés modellezése
 - Idő telik az állapotokban
 - Relatív időmérés (pl. time-out): Időzítő resetelése és leolvasása
 - Az idő függvényében változó viselkedés modellezhető
 - Ellenőrizhető: Adott időn belül (idő múlva) elérhető állapotok
- Kiterjesztés: Óraváltozók
 - Azonos rátával automatikusan „haladó” konkurens órák (időzítők)
- Használat állapotátmenetekben:
 - Akciók: Óraváltozók nullázása (resetelés), egymástól függetlenül
 - Örfeltételek: Óraváltozók és konstansok használhatók a predikátumokban
- Használat állapotokban:
 - Állapot invariánsok: Predikátum óraváltozókon és konstansokon, megadja, meddig állhat fenn az adott állapot

Az invariánsok és őrfeltételek szerepe

clock x;

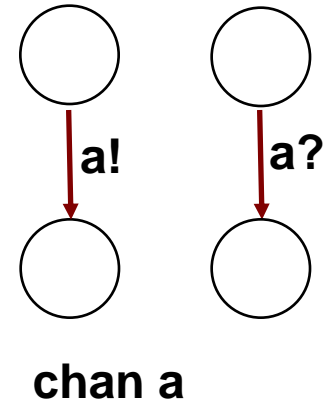


Az **open** állapot elhagyásakor a $[4, 8]$ tartományban lehet x értéke



Kiterjesztések elosztott rendszerekhez

- Cél: Együttműködő automaták hálózatának modellezése
 - Szinkronizáció az egyes automaták között
 - Együttlépő átmenetek (randevú): szinkron kommunikáció
 - Üzenet küldés és fogadás csak együtt valósulhat meg (küldő vár)
 - (Ezzel aszinkron kommunikáció is leírható)
- Kiterjesztés: Szinkronizált akciók
 - Csatornák definiálása (szinkron csatorna)
 - Üzenetküldés: **!** operátor a csatornára
 - Üzenetfogadás: **?** operátor a csatornára
 - Pl: az **a** nevű csatorna esetén **a!** és **a?** akciók
- Paraméterezés
 - Automaták paraméterezése: Példányosítás
 - Pl. **Door(bool &id)** egy **id** változó értéke lesz a paraméter
 - Paraméterezhető csatornák
 - Pl. **a[id]** egy **id** változó esetén

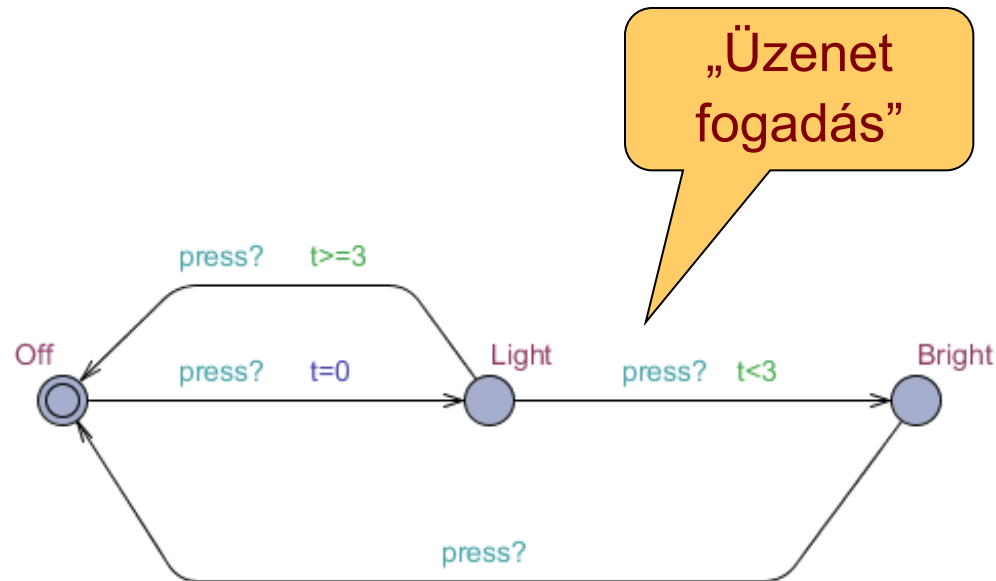


Példa óraváltozókra és szinkronizálásra

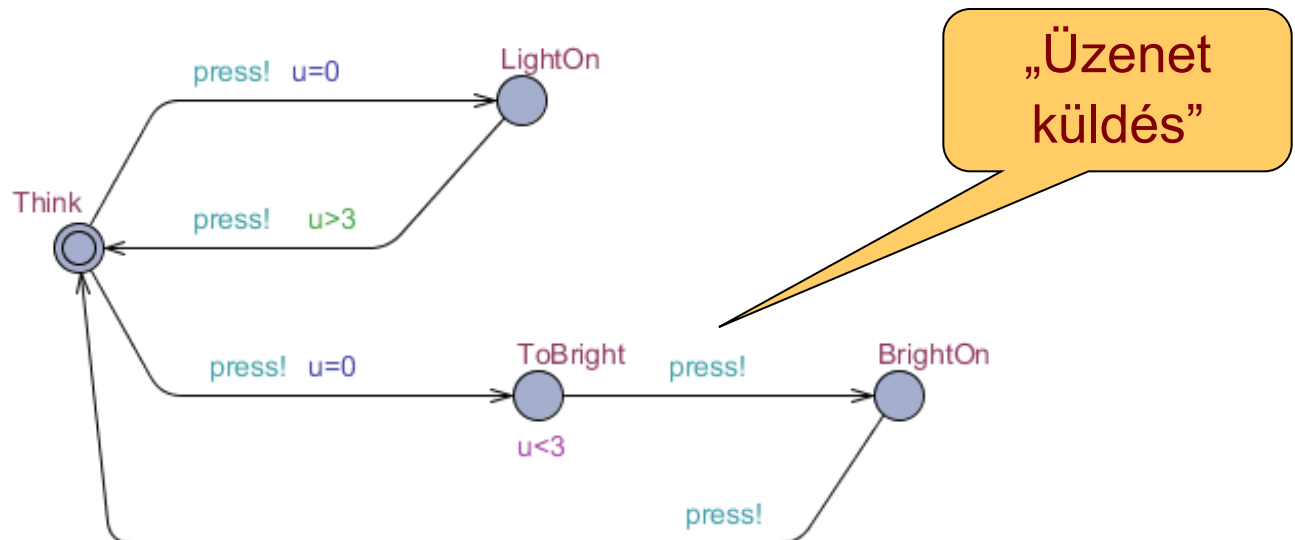
Deklarációk:

```
clock t, u;  
chan press;
```

Kapcsoló:

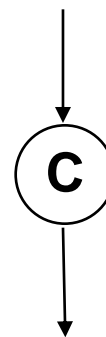


Felhasználó:

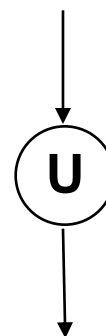


További lehetőségek: Speciális állapotok

- **Committed állapot:** átmenetek egybefogása
 - Tipikus használat: A kimenő átmenet végrehajtása előtt más automata átmenete nem lehet végrehajtva: bemenő és kimenő átmenet egy atomi műveletként végrehajtva

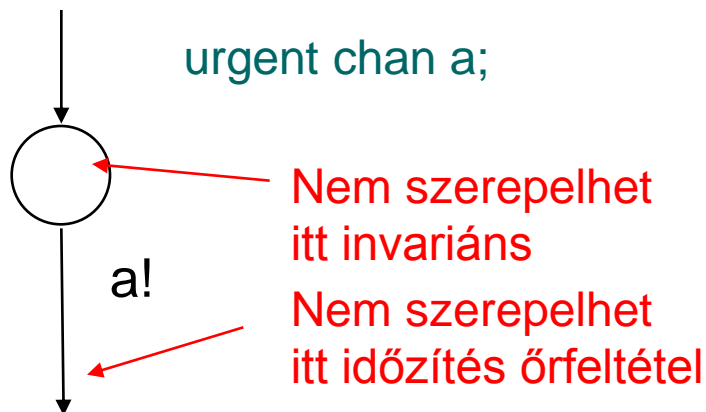


- **Urgent állapot:** késleltetés korlátozása
 - Nem telhet idő az adott állapotban, ha enélkül lehetséges a kilépés
 - Ekvivalens modell:
 - Óraváltozó bevezetése: $\text{clock } x;$
 - Minden bemenő élen resetelve: $x:=0$
 - Állapot invariáns hozzárendelése: $x \leq 0$



További lehetőségek: Urgent csatorna

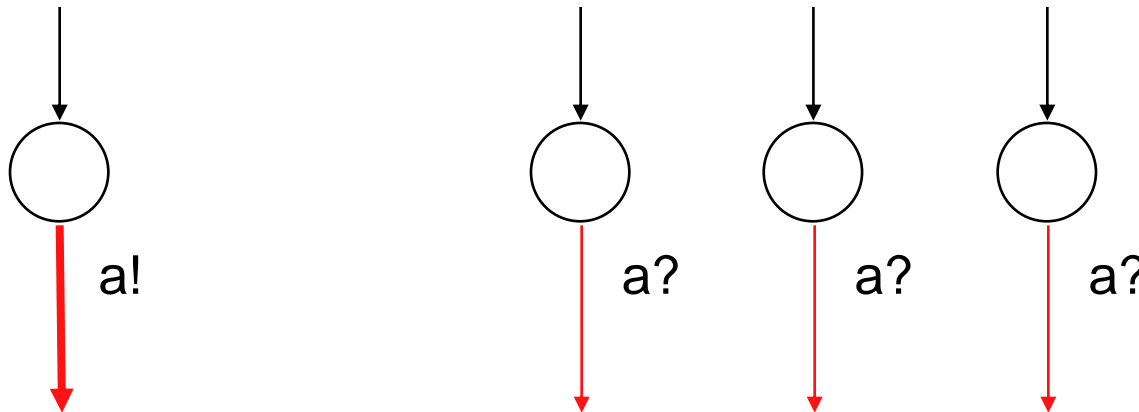
- Urgent csatorna: Nem enged késleltetést
 - Késleltetés nélkül, azonnal végrehajtandó szinkronizáció (de előtte átlapolt végrehajtás lehet)
 - Nem szerepelhet időzítés őrfeltétel azon az átmeneten, ami ilyen csatornára hivatkozó akcióval van címkézve
 - Nem szerepelhet invariáns azon a csomóponton, ahonnan olyan átmenet indul, ami ilyen csatornára hivatkozó akcióval van címkézve



További lehetőségek: Broadcast csatorna

- **Broadcast csatorna: 1->N kommunikáció**
 - „Üzenetküldés” feltétel nélkül megtörténik
 - Nem kell fogadó készenlétére (randevúra) várni
 - Minden, „üzenetfogadásra” kész partner erre szinkronizálódik
 - Üzenetfogadáshoz szükséges az üzenetküldés
 - Nem szerepelhet őrfeltétel a broadcast csatornára hivatkozó üzenetfogadó átmeneten

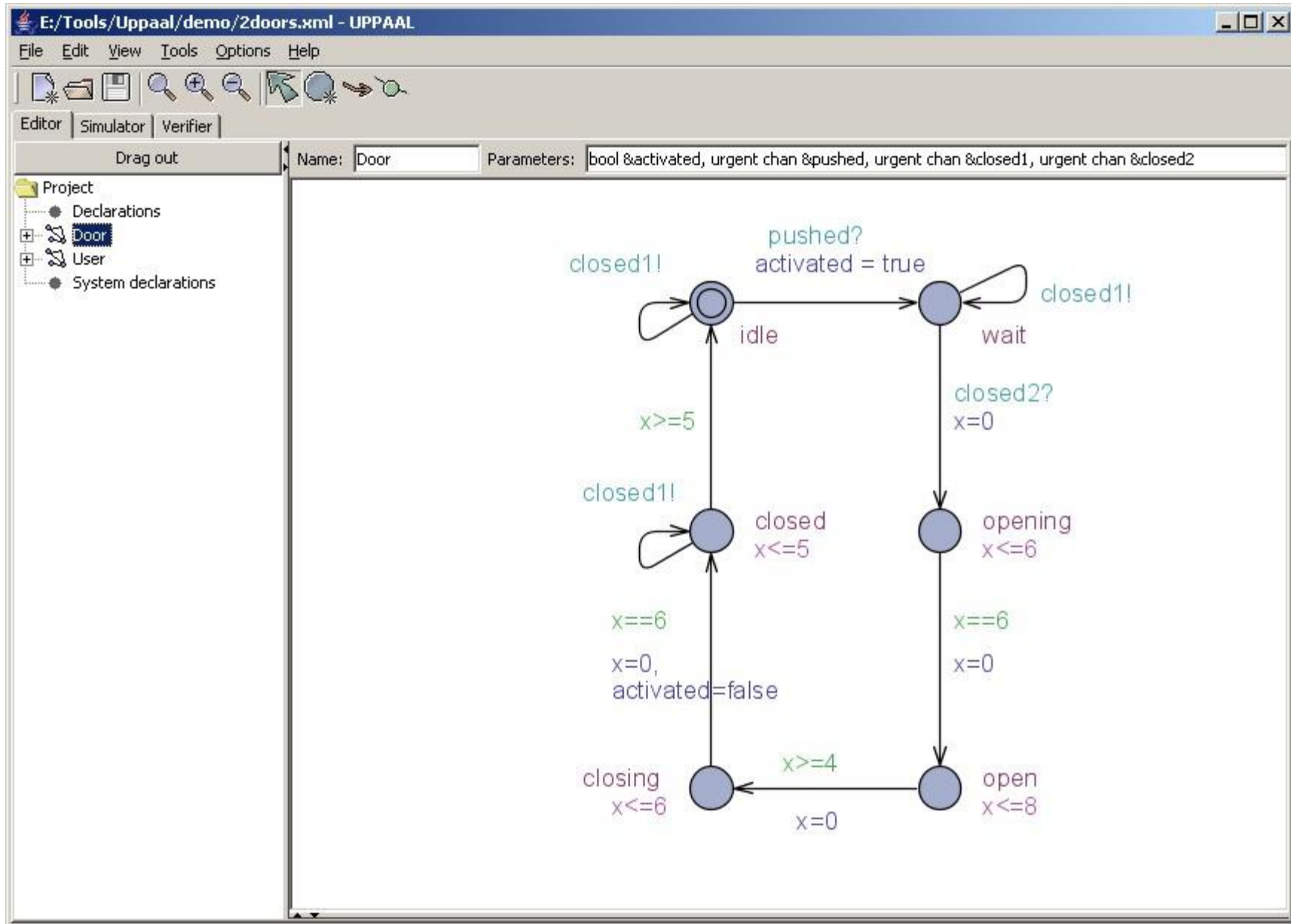
broadcast chan a;



Az UPPAAL eszköz

- Fejlesztése (1999-):
 - Uppsala University, Svédország
 - Aalborg University, Dánia
- Web lap (információk, letöltés, példák):
<http://www.uppaal.org/>
- Kapcsolódó eszközök:
 - UPPAAL CoVer: Tesztgenerálás
 - UPPAAL TRON: On-line tesztelés
 - UPPAAL PORT: Komponens alapú rendszerek tervezése
 - ...
- Kereskedelmi verzió:
<http://www.uppaal.com/>

Automata modell



Szimulátor

E:/Tools/Uppaal/demo/2doors.xml - UPPAAL

File Edit View Tools Options Help

Editor Simulator Verifier

Drag out

Enabled Transitions

User2

closed2: Door2 --> Door1

Next Reset

Simulation Trace

(idle, idle, idle, idle)

User1

(idle, idle, -, idle)

pushed1: User1 --> Door1

(wait, idle, idle, idle)

Trace File:

Prev Next Replay

Open Save Random

Slow Fast

Drag out

activated1 = 1
activated2 = 0
Door1.x >= 0
Door2.x >= 0
User1.w = 0
User2.w >= 0
Door1.x = Door2.x
Door2.x = User2.w
User2.w = Door1.x

Door1

Door2

User1

User2

Door1 Door2 User1 User2

Verifikáció

F:\FTapps\Uppaal\demo\2doors.xml - UPPAAL

File Edit View Tools Options Help

Editor Simulator Verifier

Overview

A[] not (Door1.open and Door2.open)	●
A[] (Door1.opening imply User1.w<=31) and (Door2.opening imply User2.w<=31)	●
E<> Door1.open	●
E<> Door2.open	●

Check
Insert
Remove
Comments

Query

A[] not (Door1.open and Door2.open)

Comment

Mutex: The two doors are never open at the same time.

Status

Established direct connection to local server.
(Academic) UPPAAL version 4.0.7 (rev. 4140), November 2008 -- server.
Disconnected.
Established direct connection to local server.
(Academic) UPPAAL version 4.0.7 (rev. 4140), November 2008 -- server.
A[] not (Door1.open and Door2.open)
Property is satisfied.
A[] (Door1.opening imply User1.w<=31) and (Door2.opening imply User2.w<=31)
Property is satisfied.
E<> Door2.open
Property is satisfied.
A[] not deadlock
Property is satisfied.
Door2.wait --> Door2.open
Property is satisfied.
Door1.wait --> Door1.open
Property is satisfied.