Operációs rendszerek (vimia219)

Feladatok együttműködésének ellenőrzése

Micskei Zoltán

http://www.mit.bme.hu/~micskeiz

Méréstechnika és Információs Rendszerek Tanszék

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Utolsó módosítás: 2012. 03. 20.

Az előadás a R. Hamberg and F. Vaandrager. Using Model Checkers in an Introductory Course on Operating Systems. OSR 42(6):101-111. cikkben közzétett modelleket használja fel. (URL: http://www.mbsd.cs.ru.nl/publications/papers/fvaan/MCinEdu/)

Hyman algoritmusa

```
turn=0, flag[0]=flag[1]=false;

Protocol (int id) {
  do {
      flag[id] = true ;
      while (turn != id) {
            while (flag[1-id]) /* do nothing */ ;
            turn = id;
      }
      CriticalSection(id);
      flag[id] = false;
  } while (true) ;
}
```

Lehetnek-e ketten egyszerre a kritikus szakaszban?

Harris Hyman, Comments on a problem in concurrent programming control, Communications of the ACM, v.9 n.1, p.45, Jan. 1966

Peterson algoritmusa

```
turn=0, flag[0]=flag[1]=false;

Protocol (int pid) {
   while (true) {
     flag[pid]=true;
     turn=1-pid;
     while (flag[1-pid]&&turn==1-pid) /**/;
     CriticalSection(id);
     flag[pid]=false;
   }
}
```

Lehetnek-e ketten egyszerre a kritikus szakaszban?

G.Peterson. Myths about the mutual exclusion problem. Inf. Process. Lett., 12(3):115–116, 1981.

Algoritmusok helyességének ellenőrzése

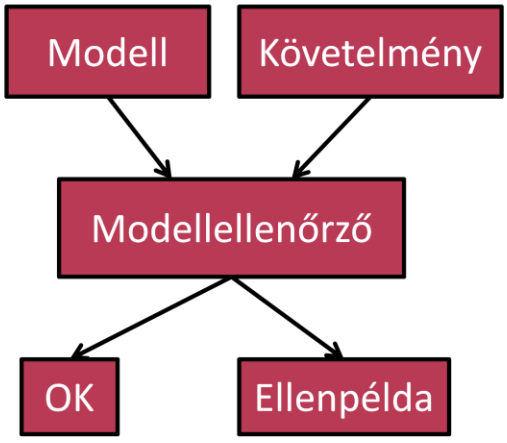
- Hogyan döntsük el, hogy jó?
- Erősen nézzük, és próbálunk rájönni:)
- Végigpróbálunk néhány lefutást
  - Ha hibázik: javítjuk a kódot
  - Ha nem találunk hibát: ??
- Szisztematikus megoldás kell:
  - „formális módszerek”

Algoritmusok helyességének ellenőrzése

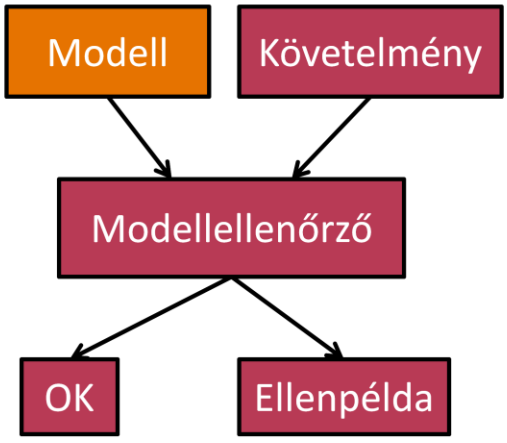
- Milyen jó lenne egy eszköz:
  - Algoritmusaink egyszerű **leírására**
  - Rendszer működésének **szimulálására**
  - Összetett követelmények **megfogalmazására**
  - Követelmények **ellenőrzésére** gombnyomásra
- Jó hír: vannak ilyen eszközök☺
  - **Modellellenőrzők** (model checkers)
  - 30+ év kutatás eredménye
  - Valós ipari eredmények HW és SW rendszereknél

ACM Turing Award Honors Founders of Automatic Verification Technology
http://www.acm.org/press-room/news-releases-2008/turing-award-07/

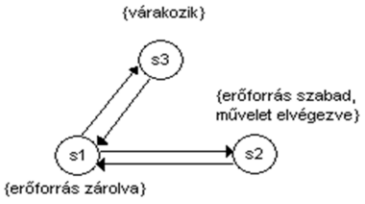
Modellellenőrzők



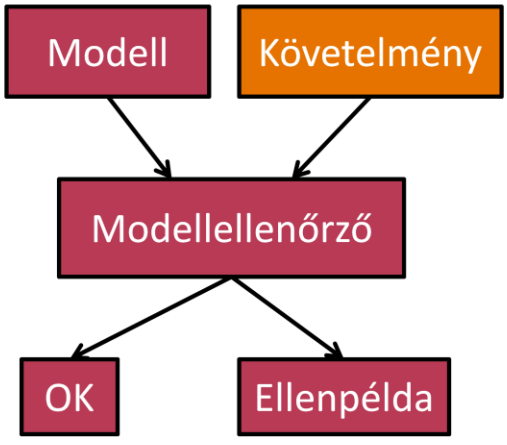
Modellellenőrzők



- Rendszer működésének leírása
- Tipikusan valami állapotgépszerű



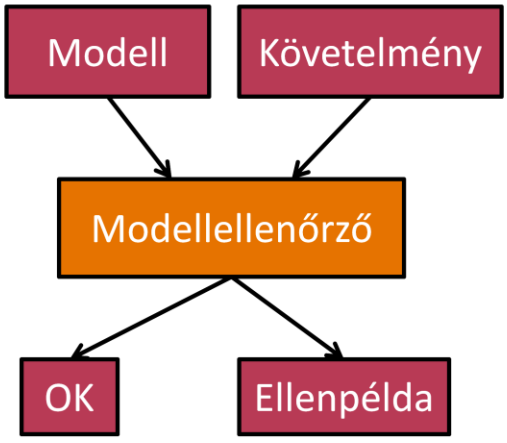
Modellellenőrzők



- Mit akarunk ellenőrizni
  - Kölcsönös kizárás
  - Holtpont mentesség
  - …
- Logikai kifejezés:
  - Pl.:

  ! (A_var AND B_var)

Modellellenőrzők

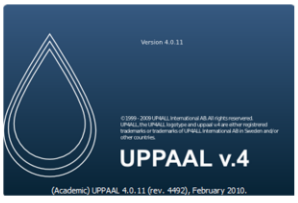


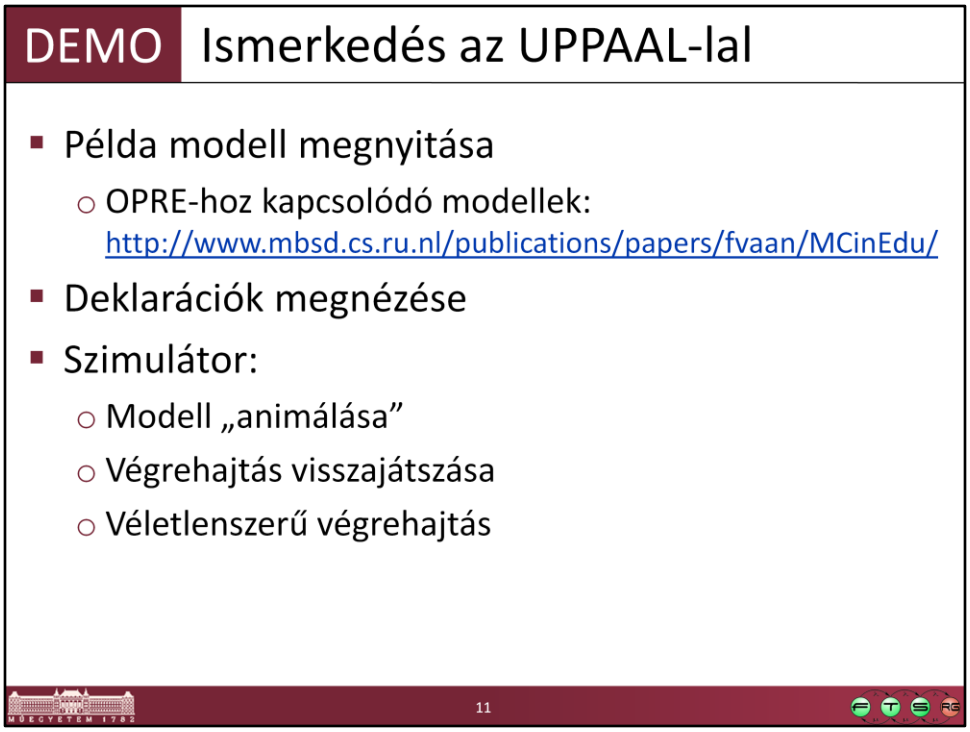
- „Fekete doboz”
- Automatikus

- Eredmény:
  - Követelmény igaz
  - Követelmény nem teljesül + ellenpélda

UPPAAL

- Időzítést is támogató modellellenőrző
- Uppsala & Aalborg egyetemek, 15+ éve fejlesztik
- Cél: hatékonyság, könnyű használhatóság
- http://www.uppaal.com/
  - Akadémiai célra ingyenesen letölthető
  - Leírások
  - Részletes súgó
  - Esettanulmányok
  - Sok kiegészítés (tesztgenerálás)

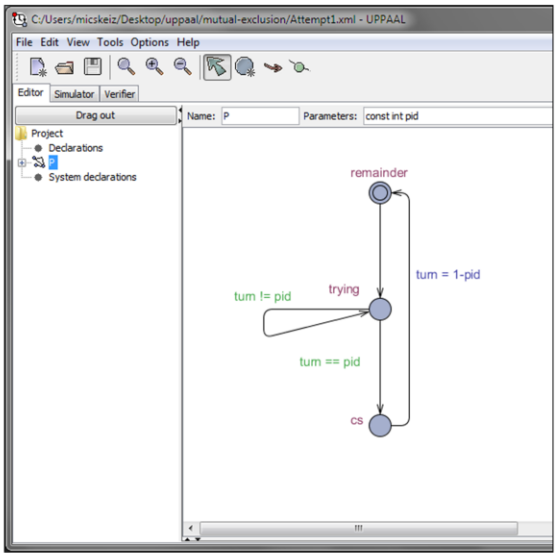




Nézzük meg valamelyiket a kölcsönös kizárás (mutual exclusion) példák közül. Miért lesz nekünk jó ez:

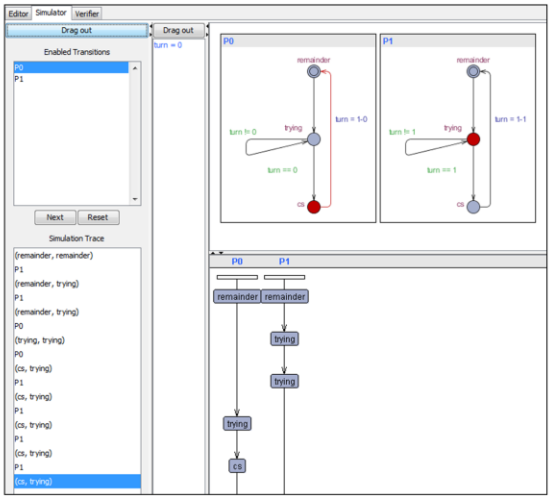
- Le tudjuk írni az algoritmusunkat, a grafikus forma talán elsőre könnyebben érthető.
- A szimulációnál könnyen végig tudjuk próbálni, hogy tényleg úgy működik-e, ahogyan képzeltük.
- Meg tudjuk jeleníteni a futást, azt el is tudjuk menteni. Később például egy problémás esetet egyszerűen meg tudunk másnak is mutatni.
- Véletlenszerű végrehajtással előállhat könnyen olyan eset, amire nem is gondoltunk.
- …

Az UPPAAL felülete: modell szerkesztő



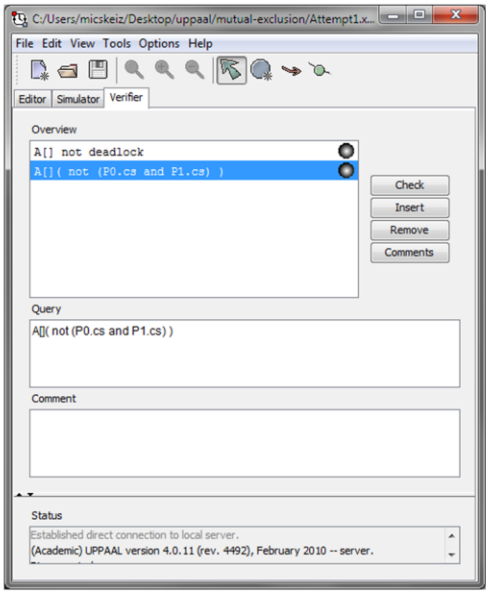
- Globális változók
- Automata
  - Állapot
  - Átmenet
    - Őrfeltétel
    - Akció
  - Órák
- Rendszer:
  - Automata példányok

Az UPPAAL felülete: szimulátor



- Átmenet kiválasztása
- Változók állapota
- Automaták képe
- Trace:
  - Szöveges
  - Grafikus: *Message Sequence Charts*

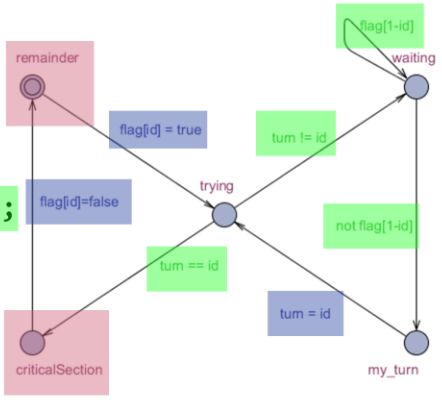
Az UPPAAL felülete: ellenőrzés



- Követelmény:
  - Logikai formula
- Elemei:
  - Állapotra hivatkozás
  - NOT, AND, OR
- További operátorok:
  - A: minden úton
  - E: legalább egy úton
  - [ ]: minden időben
  - <>: valamikor a jövőben

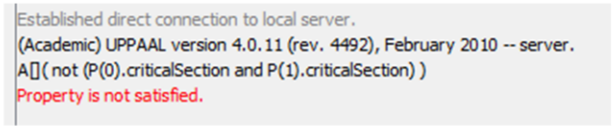
Vissza a Hyman algoritmushoz

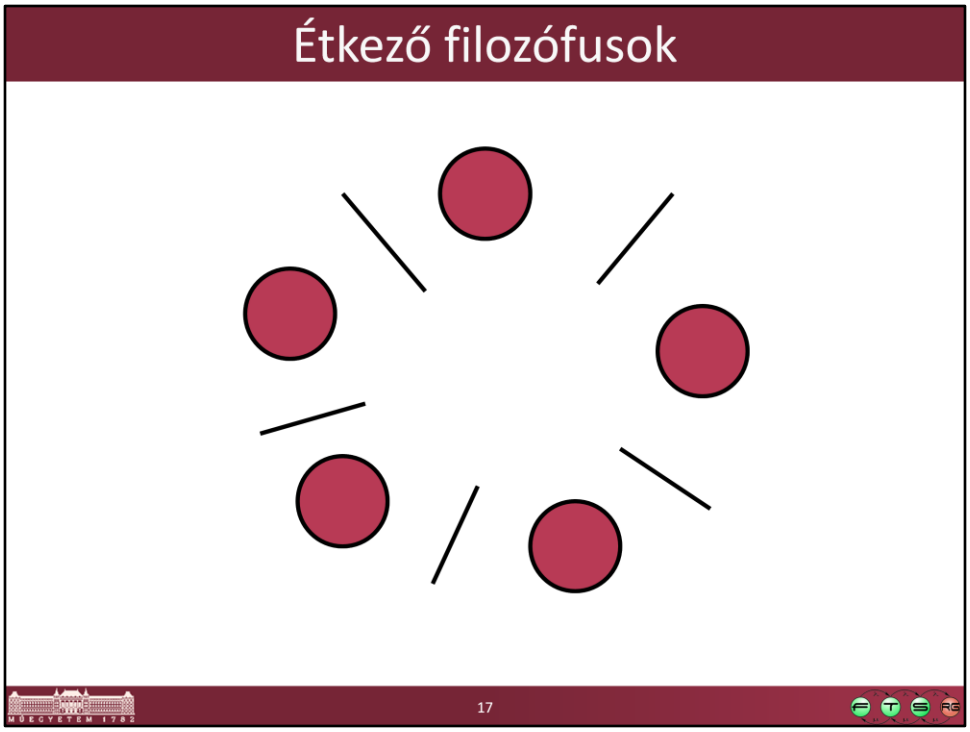
```
turn=0, flag[0]=flag[1]=false;

Protocol (int id) {
   do {
      flag[id] = true ;
      while (turn != id) {
            while (flag[1-id]);
            turn = id;
      }
      CriticalSection(id);
      flag[id] = false;
   } while (true) ;
}
```



DEMO Hyman és Peterson algoritmusa

- Algoritmusokat leíró modellek vizsgálata
- Szimuláció
- Követelmények ellenőrzése:
  - Egyszerre csak egy példány lehet a kritikus szakaszban:
  - **A[](not (P(0).criticalSection and P(1).criticalSection))**
- Ellenpélda generálása:
  - *Options / Diagnostic Trace / Shortest*

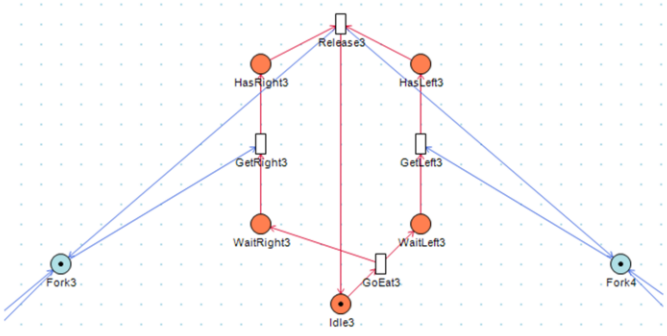




Filozófusok ülnek az asztal körül, és az evéshez a bal és jobb oldalon lévő pálcika felvételére is szükség van.

DEMO Holtpont – Étkező filozófusok

- petridotnet from BME-MIT - Tanszéki fejlesztés, TDK díjas
- Modellek: Petri háló
- Sokféle analízis lehetőség



PetriDotNet: http://petridotnet.inf.mit.bme.hu/

További eszközök

- Java Pathfinder
  - Modellellenőrző Java byte kódhoz
  - NASA fejlesztés, 2005 óta nyílt forrású
- CHESS
  - .NET-es kódokhoz
  - Párhuzamosságból fakadó hibák keresése
- jchord
  - Java kód statikus analízise
  - Versenyhelyzet, holtpont detektálás
- Static Driver Verifier (SDV, korábban SLAM)
  - Windows eszközmeghajtók ellenőrzése
- ...

- Java Pathfinder, http://babelfish.arc.nasa.gov/trac/jpf/wiki
- CHESS, http://research.microsoft.com/en-us/projects/chess/
- jchord, http://code.google.com/p/jchord/
- SDV-ről egy összefoglaló és példák a használatára (egy korábbi opre fakultatív feladat)
  - Ferencz Endre. Static Driver Verifier, 2010., http://mit.bme.hu/~micskeiz/opre/files/opre-sdv.pdf

Összefoglalás

- Feladatok együttműködésénél sok hibalehetőség
- Versenyhelyzet, holtpont…
- DE: léteznek eszközök a vizsgálathoz
- Modellellenőrzők, tételbizonyítók, statikus ellenőrzők…

További információ

- R. Hamberg and F. Vaandrager. Using Model Checkers in an Introductory Course on Operating Systems. OSR 42(6):101-111.
- Formális módszerek MSc tantárgy (VIMIM100)
- UPPAAL modellellenőrző
- PetriDotNet modellellenőrző

- Formális módszerek, http://www.inf.mit.bme.hu/edu/courses/form
- UPPAAL, http://www.uppaal.com/
- PetriDotNet, http://petridotnet.inf.mit.bme.hu/