Gyakorló feladatok: Formális modellek, temporális logikák, modellellenőrzés

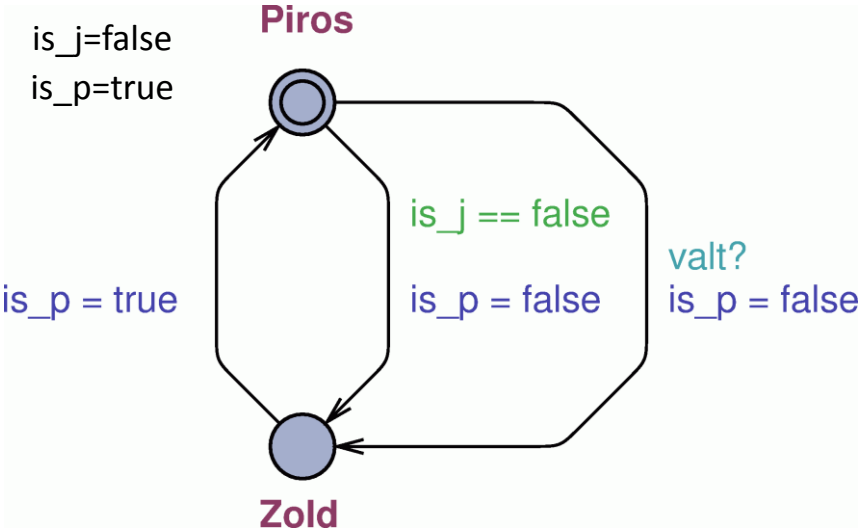
Majzik István
BME Méréstechnika és Információs Rendszerek Tanszék

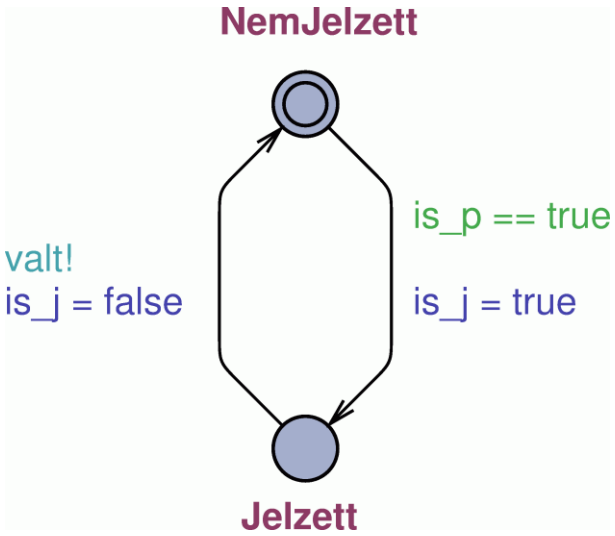
Formális modellek használata és értelmezése

Formális modellek értelmezése

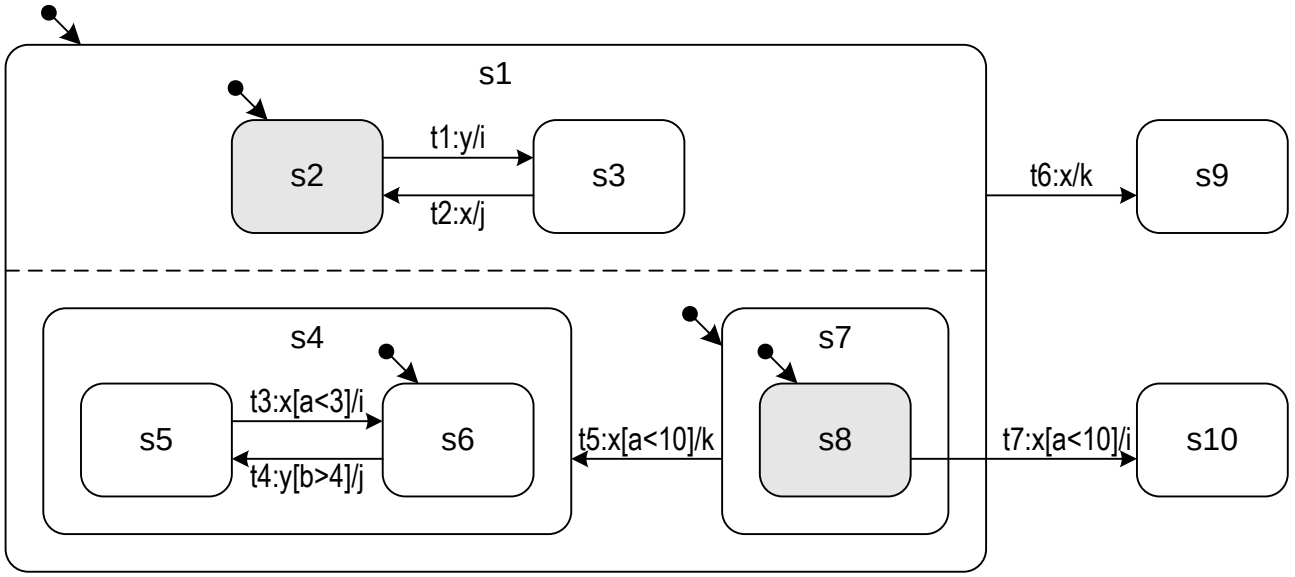
Az alábbi ábrákon látható két (az UPPAAL eszközben felvett) automata, ezek egy jelzőlámpa és egy gyalogos viselkedését modellezik. A kezdeti állapotban is_p=true, is_j=false.

- Készítse el a két automata együtteseként tekintett teljes rendszer Kripke-struktúra modelljét, a jelzőlámpa és a gyalogos elérhető állapotkombinációit és a köztük lévő átmeneteket felvéve. A Kripke-struktúra minden állapotát címkézze meg azzal, hogy a jelzőlámpa és a gyalogos mely állapotait reprezentálja.





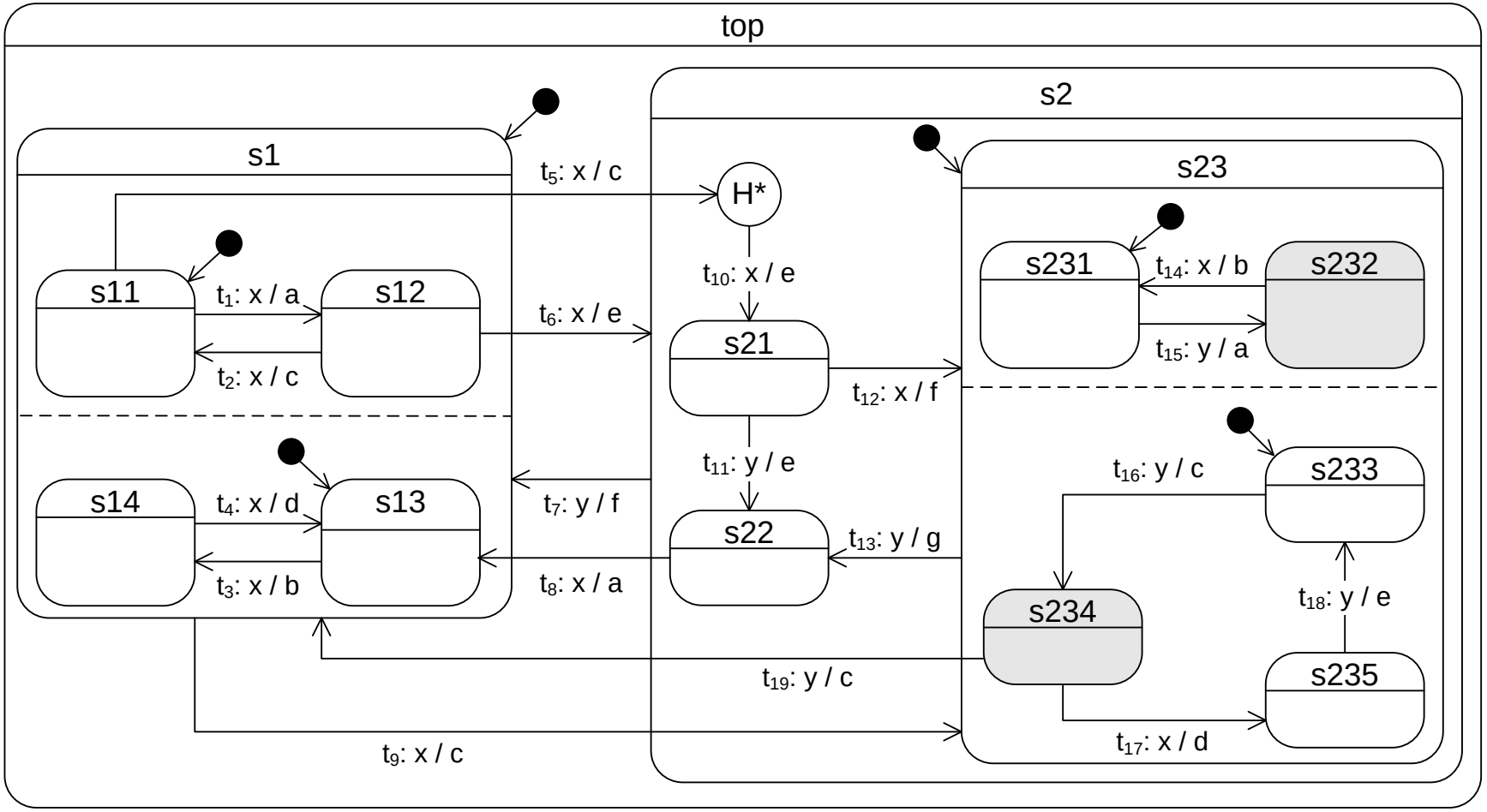
Állapottérképek 1.



A kezdő állapotkonfigurációban az *a* változó értéke 8 és egy „*x*” esemény érkezik.

1. Melyek az engedélyezett állapotátmenetek?
2. Mely engedélyezett állapotátmenetek vannak konfliktusban?
3. Hogy néz ki a tüzelhető állapotátmenetek halmaza?
4. Hogy néz(nek) ki a következő stabil állapotkonfiguráció(k)?
5. Milyen akciók és milyen sorrendben hajtódnak végre?

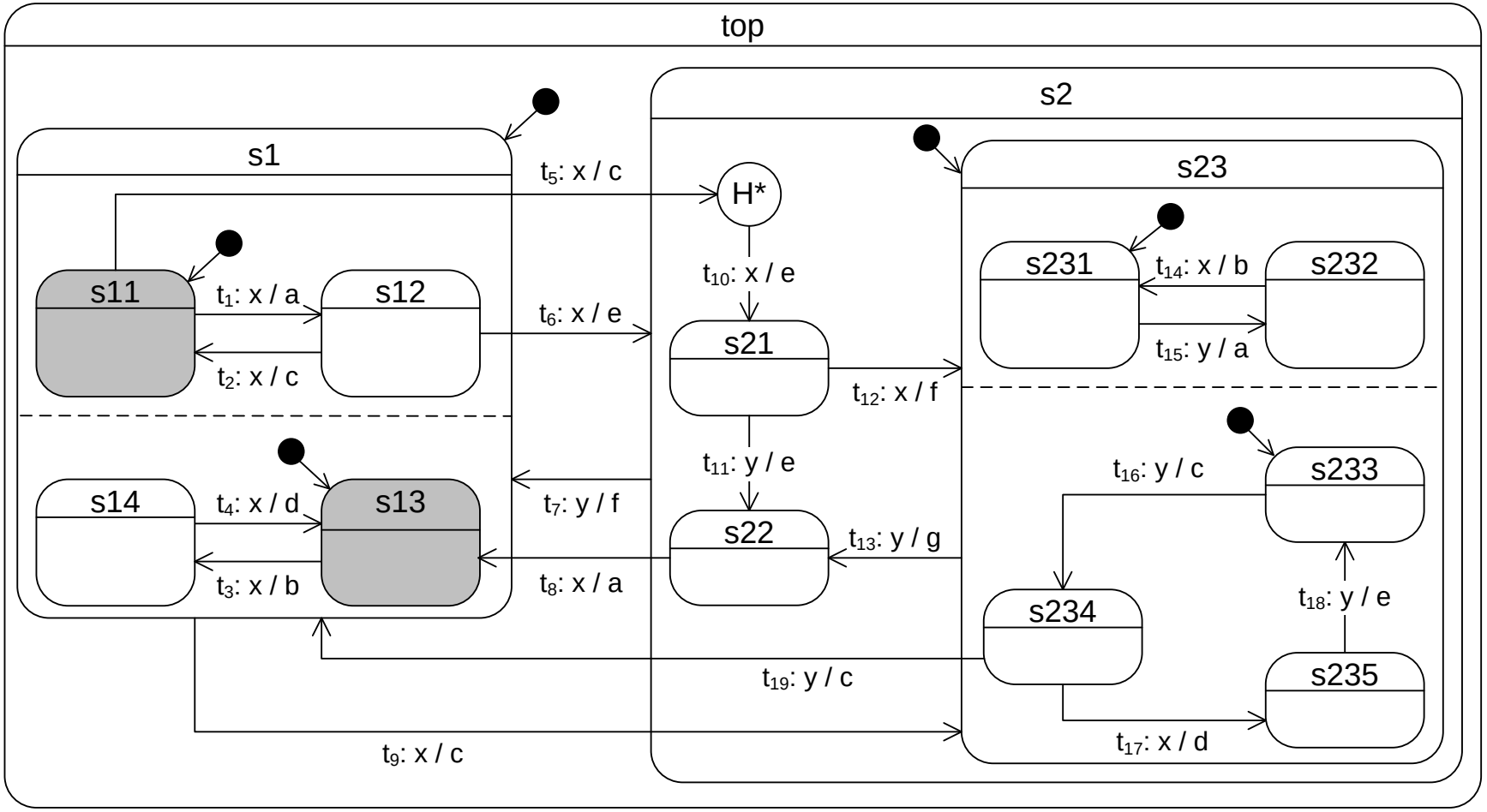
Állapottérképek 2.1.



A {top, s2, s23, s232, s234} állapotkonfigurációban az *y* esemény érkezik az eseménykezelőtől.

- Mi lesz az új állapotkonfiguráció?

Állapottérképek 2.2.



Az így elért (az ábrán szereplő) állapotban az *x* esemény érkezik.

1. Melyek az engedélyezett, a konfliktusban lévő, illetve a tüzelő átmenetek?
2. Mi lesz az új állapotkonfiguráció? Milyen akciók hajtódnak végre?

Követelmények formalizálása temporális logikákkal

Temporális logikai kifejezések értelmezése

Indokolja meg, hogy következő LTL ekvivalencia helyes-e:

1. (F Stop) $\vee$ (F Start) $\equiv$ F (Stop $\vee$ Start)
2. G Stop $\equiv$ not F (not Stop)

Indokolja meg, hogy következő CTL ekvivalencia helyes-e:

1. AF (Start $\vee$ Stop) $\equiv$ (AF Start) $\vee$ (AF Stop)
2. AF (Start $\wedge$ Stop) $\equiv$ (AF Start) $\wedge$ (AF Stop)
3. EF (Start $\wedge$ Stop) $\equiv$ (EF Start) $\wedge$ (EF Stop)

Indokolja meg, hogy az alábbi kifejezés szintaktikailag helyes-e CTL illetve CTL* temporális logikában!

1. A (X Stop $\vee$ F Start)
2. A (Stop U (AX Start))

Követelményformalizálás: Vasúti kereszteződés

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

Követelményformalizálás: Szerverterem

- 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: {készenlét, normál, maximális}
- Formalizálja LTL kifejezések segítségével az alábbi követelményeket, amelyek a rendszer működésére minden esetben vonatkoznak:
  1. Ha egy adott pillanatban a szimuláció a hűtőberendezés készenléti állapota mellett zajlik, akkor a következő pillanatban a szerver várakozó állapotra kapcsol.
  2. Előbb-utóbb elkezdhető a szimuláció.
  3. Csak úgy hajtható végre szimuláció, ha volt bemelegítés a hűtőberendezés normál működése mellett.

Tulajdonságok ellenőrzése formális modelleken

Modellellenőrző algoritmusok alapjai

1. Rajzolja fel a tabló felbontás szabályát a PLTL temporális logika U operátora esetén!
   Írja le, mikor adódhat ellentmondásos ág az U operátorral felírt kifejezés így megadott felbontásának elvégzése során!
2. Írja le, hogyan azonosíthatók azok az állapotok a modellben, amelyeken igaz az E(P U Q) tulajdonság!
3. Írja le, milyen átalakítási lépésekkel kapunk ROBDD-t egy bináris döntési fából!
4. Írja le a korlátos modellellenőrzés alapötletét!

Modellellenőrzés: Szerverek

- Egy informatikai rendszer egy adatbázisszerverből és egy alkalmazásszerverből áll, amelyek kikapcsolt vagy bekapcsolt állapotban lehetnek. Alaphelyzetben mindkét szerver ki van kapcsolva.
- A szervereket hibamentes esetben egyszerre kapcsolják ki/be.
- Az üzemállapot az, amikor mindkét erőforrás be van kapcsolva.
- Ha az üzemállapotban az adatbázisszervert hiba következtében kikapcsolják, az rendszerszinten üzemképtelen állapotnak tekinthető. Ezután az alkalmazásszervert is kikapcsolják, majd mindkét erőforrás bekapcsolásával indítják újra a rendszert.
- Feladatok:
  1. 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, üzemállapot, üzemképtelen}
  2. Ellenőrizze a modellen, hogy az üzemállapotból tekintve teljesül-e a következő CTL kifejezés:
     E(¬üzemképtelen U alaphelyzet)

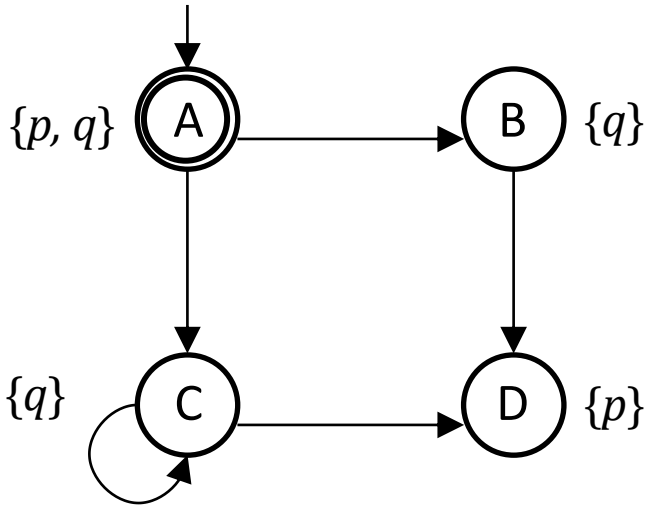
Modellellenőrzés: Informatikus hallgató

- Egy informatikus hallgató „állapotait” az alapján különböztetjük meg, hogy kávézik vagy nem, valamint 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 vizsgázást követően csak pihenés után tanul.
- Feladatok:
  1. 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}
  2. Ellenőrizze a modellen, hogy a hallgató alapállapotából (ami a tanulás) kiindulva teljesül-e a következő CTL kifejezés:
     E(¬vizsgázik U pihen)

CTL tulajdonság ellenőrzése címkézéssel

Adott az alábbi Kripke-struktúra.

- A tanult iteratív állapotcímkézési eljárást végrehajtva ellenőrizze a modellen, hogy teljesül-e a kezdőállapotból az alábbi CTL kifejezés: **A**(p **U** (**EX** ¬q)).
  Az iteráció minden lépéséhez adja meg a címkéző kifejezést és (felsorolással) a címkézett állapotok halmazát!

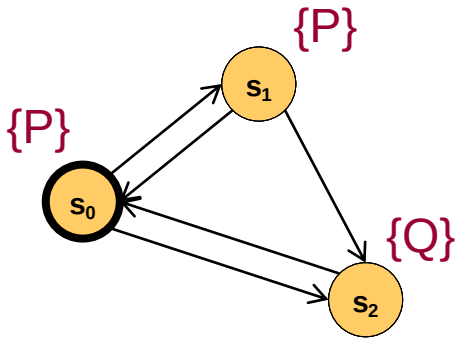


Modellellenőrzés tabló módszerrel (1)

Adott a rajzon látható Kripke struktúra.

Végezzük el a következő kifejezés ellenőrzését a tabló módszert alkalmazva:

$\neg$ (P U Q)

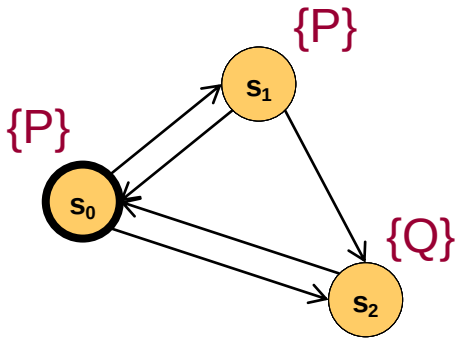


Modellellenőrzés tabló módszerrel (2)

Adott a rajzon látható Kripke struktúra.

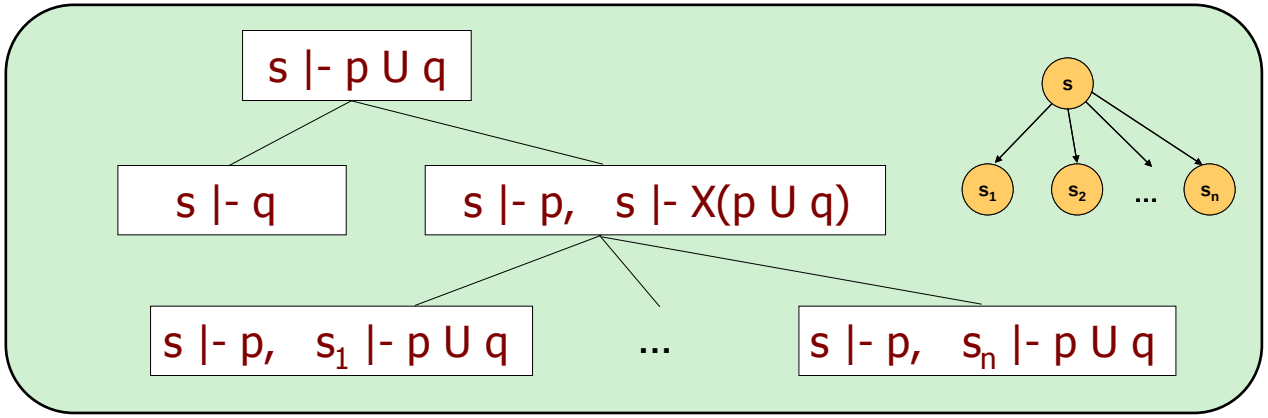
Végezzük el a következő kifejezés ellenőrzését a tabló módszert alkalmazva:

$\neg$ (P U Q)



Tudnivalók:

- Negált kifejezés (ellenpélda kereséshez): (P U Q)
- Tabló szabálya: (P U Q) = Q $\vee$ (P $\wedge$ X(P U Q))

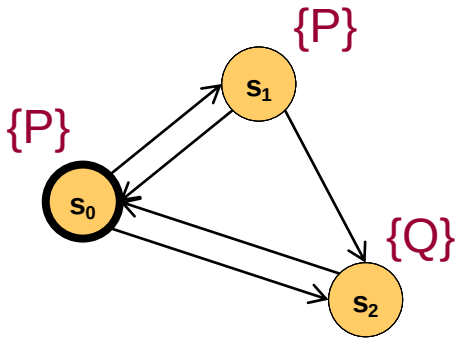


Modellellenőrzés tabló módszerrel (3)

Adott a rajzon látható Kripke struktúra.

Végezzük el a következő kifejezés ellenőrzését a tabló módszert alkalmazva:

$$\neg (P \; U \; Q)$$



Tudnivalók:

- Negált kifejezés (ellenpélda kereséshez): $(P \; U \; Q)$
- Tabló szabálya: $(P \; U \; Q) = Q \vee (P \wedge X(P \; U \; Q))$
- A tabló építésben ellentmondásra jutunk:
  - Atomi kijelentésre vonatkozó lokális állítás nem teljesül
  - X operátor van, de az útvonal véget ér Q teljesülése nélkül
  - Ciklus alakul ki P teljesülésével, de Q teljesülése nélkül
- A tabló sikeres ágai (itt ellenpéldát adnak):
  - Atomi kijelentésekre vonatkozó állítások listája teljesül
  - Ciklus alakul ki ellentmondás nélkül

ROBDD kézi összeállítása

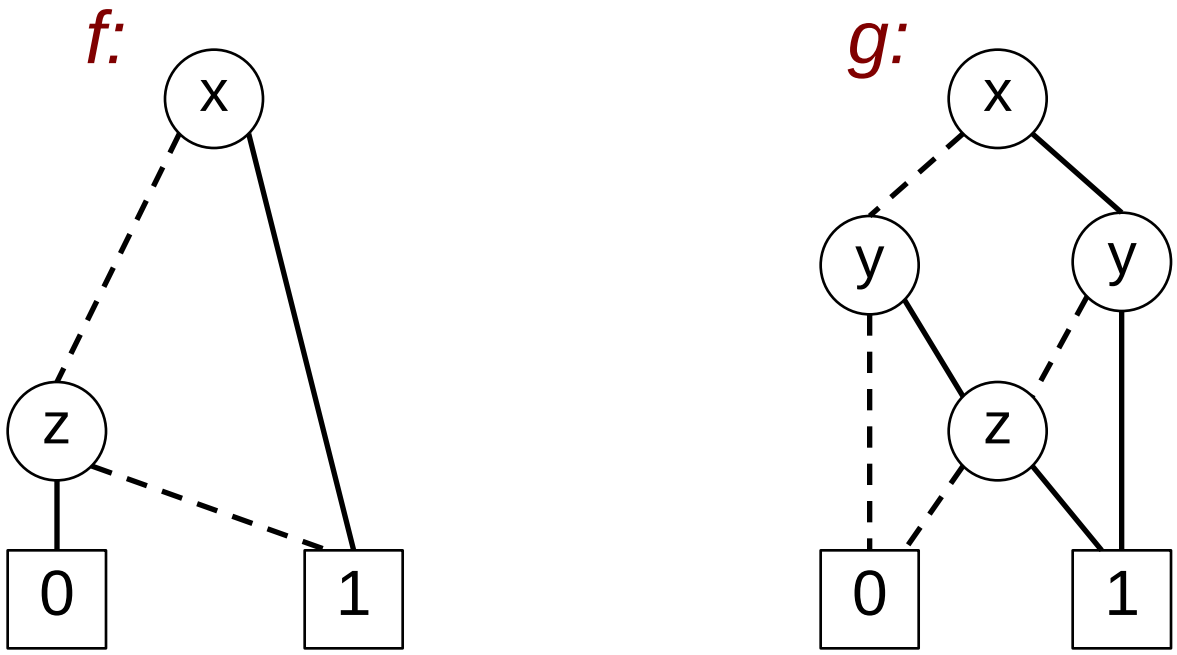
Adott a *g* logikai függvény igazságtáblázata:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1. Rajzolja fel a *g* logikai függvény döntési fáját!
   A rajzoláshoz az *x*, *y*, *z* változósorrendet használja.
2. Ez alapján adja meg a *g* függvényt redukált rendezett bináris döntési diagram (ROBDD) alakban!
3. Adja meg a függvényt algebrai (képlet) alakban!

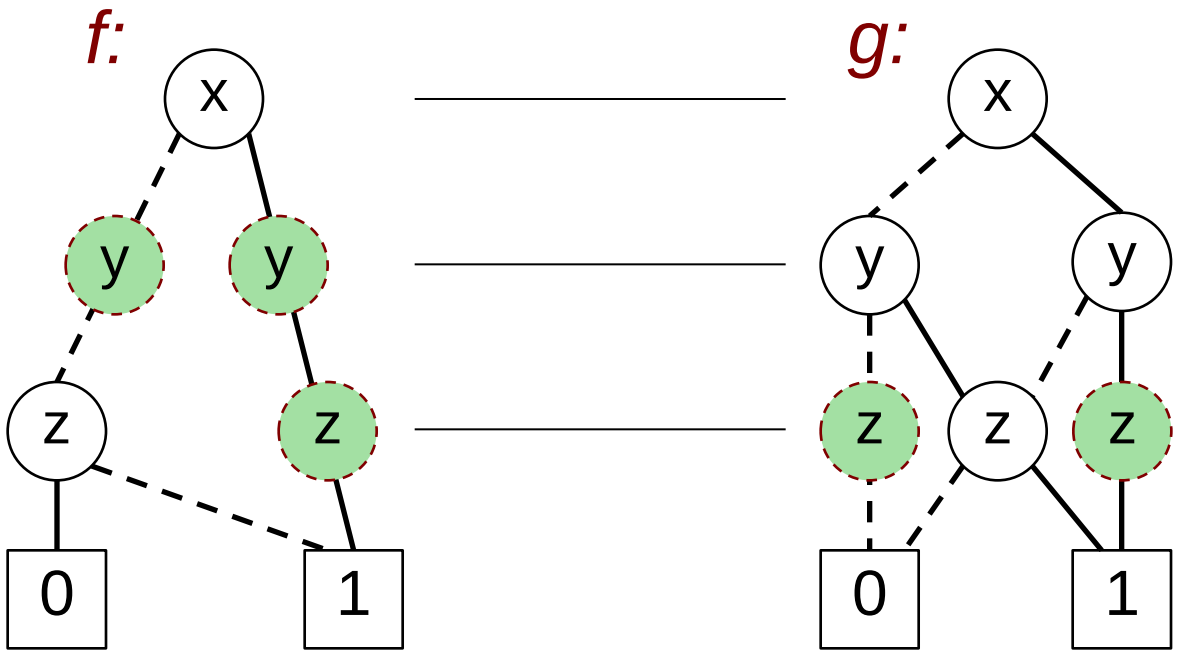
ROBDD alapú műveletek függvényeken (1)

Tekintse az alábbi, ROBDD alakban megadott *f* és *g* függvényeket, és rajzolja fel ezek alapján az $f \wedge g$ függvényt ROBDD alakban!



ROBDD alapú műveletek függvényeken (2)

Tekintse az alábbi, ROBDD alakban megadott *f* és *g* függvényeket, és rajzolja fel ezek alapján az $f \wedge g$ függvényt ROBDD alakban!



ROBDD alapú műveletek függvényeken (3)

Tekintse az alábbi, ROBDD alakban megadott *f* és *g* függvényeket, és rajzolja fel ezek alapján az $f \wedge g$ függvényt ROBDD alakban!