Formális módszerek

Előadó:
Gombás Éva, adjunktus
gombas@inf.u-szeged.hu
Számítástudomány Alapjai Tanszék
Irinyi épület 3. lépcsőház, 1. em.
Fogadóóra: csütörtök 13-14 között

Tárgyleírás:

A tárgy oktatásának célja (kialakítandó kompetenciák):

Napjaink informatikai rendszerei eddig soha nem látott komplexitást értek el, és ezzel együtt növekszik a megbízható működésük iránti igény is. Ezért egyre inkább előtérbe kerülnek azok a matematikai módszerek, melyek segítséget nyújtanak az informatikai rendszerek tervezésében és ellenőrzésében, meghatározó szerepet vállalva a kívánt megbízhatóság elérésében.

A kurzus ezeket az úgynevezett formális módszereket kívánja bemutatni, ismertetve nemcsak a matematikai hátteret, hanem a használatukat megkönnyítő szoftver eszközöket is.

Tematika:

- A formális módszerek áttekintése és szerepük.
- Átmeneti rendszerek, időzített automaták, időzített automaták hálózatai.
  Petri hálók.
  Processzus algebrák.
- Formális specifikáció. Temporális logikák, lineáris (LTL) és elágazó idejű logikák (CTL, CTL*), időzített logikák (TCTL).
- Modell-ellenőrzés, feladata és menete. Modell-ellenőrzési algoritmusok. Szimbolikus modell-ellenőrzés, bináris döntési diagrammok. Modell-ellenőrző eszközök (Spin, UPPAAL).

Ajánlott irodalom:

- 1. Pataricza A. (szerk): Formális módszerek az informatikában, 2. kiadás, Typotex, 2005.
- 2. Ésik Z., Gombás É., Németh L. Z.: Hardver és szoftver rendszerek verifikációja, Typotex Kiadó, 2011.
- 3. L. Aceto, A. Ingólfsdóttir, K. G. Larsen, J. Srba, Reactive Systems: Modelling, Specification and Verification, Cambridge University Press, 2007.
- 4. A. Arnold: Finite Transition Systems: Semantics of Communicating Systems, Prentice-Hall International Series in Dynamics, Prentice-Hall, 1994.
- 5. Ch. Baier and J--P. Katoen, Principles of Model Checking, MIT Press, 2008.
- 6. G. Behrmann, A. David, K. G. Larsen: A Tutorial on Uppaal, in: Proc. SFM-RT'04, LNCS 3185, 2004
- 7. B. Berard, M. Bidiot, A. Finkel, F. Laorussinie, A. Petitt, L. Pterucci, Ph. Schnoebelen: Systems and Software Verification, Springer, 2001.
- 8. G. J. Holzmann, The SPIN Model Checker: Primer and Reference Manual, Addison-Wesley, 2003.

Ajánlott szoftverek:

- UPPAAL: http://www.uppaal.org/
- jSpin: http://code.google.com/p/jspin/
- Spin: http://spinroot.com/
- SAL: http://sal.csl.sri.com/
- CPN Tools: http://cpntools.org/
- LTSA: http://www.doc.ic.ac.uk/ltsa/

A vizsga (kollokvium) teljesítésének feltételei:

- Csak az a hallgató vizsgázhat az ETR-ben meghirdetett vizsganapokon, aki a gyakorlatot teljesítette, érvényes vizsgajelentkezése van az adott vizsganapra az ETR-ben és igazolja személyazonosságát (a leckekönyv /diákigazolvány bemutatásával.
- Szóbeli vizsgán az előre kiadott, két részből álló tételsor mind a két részéből a hallgató egy-egy tételt húz.
- A szóbeli feleletre a felkészülési idő 30 perc.
- A kollokvium jegye a két tételre adott szóbeli felelet alapján alakul ki. Ha az egyik tételből a felelet elégtelen, akkor a teljes vizsga is sikertelen.

- A feleltetéskor a húzott tételeken kívül a tananyag más részére is rákérdezhet a vizsgáztató.
  Bizonyos alapkérdések ismeretének hiánya esetén a vizsgáztató elégtelenre minősítheti a vizsgát, függetlenül a többi kérdésre adott választól.
- Írásbeli vizsga esetén a maximális megszerezhető pontszám 40, amely előre kiadott tételsorból az oktató által kijelölt két, egyenként 20 pontos tétel kidolgozásával szerezhető meg. A vizsga akkor sikeres, ha mind a két tétel kidolgozása minimum 6 pontos. Sikeres vizsga érdemjegye a tételekre kapott pontszám összege alapján az alábbiak szerint történik:
  0 - 11 pont : elégtelen (1)
  12 - 18 pont : elégséges (2)
  19 - 25 pont : közepes (3)
  26 - 32 pont : jó (4)
  33 - 40 pont : jeles (5)

2015-2016. tanév 2. félévében a vizsga szóban lesz.

Verifikáció (ellenőrzés)

Miért van rá szükség?

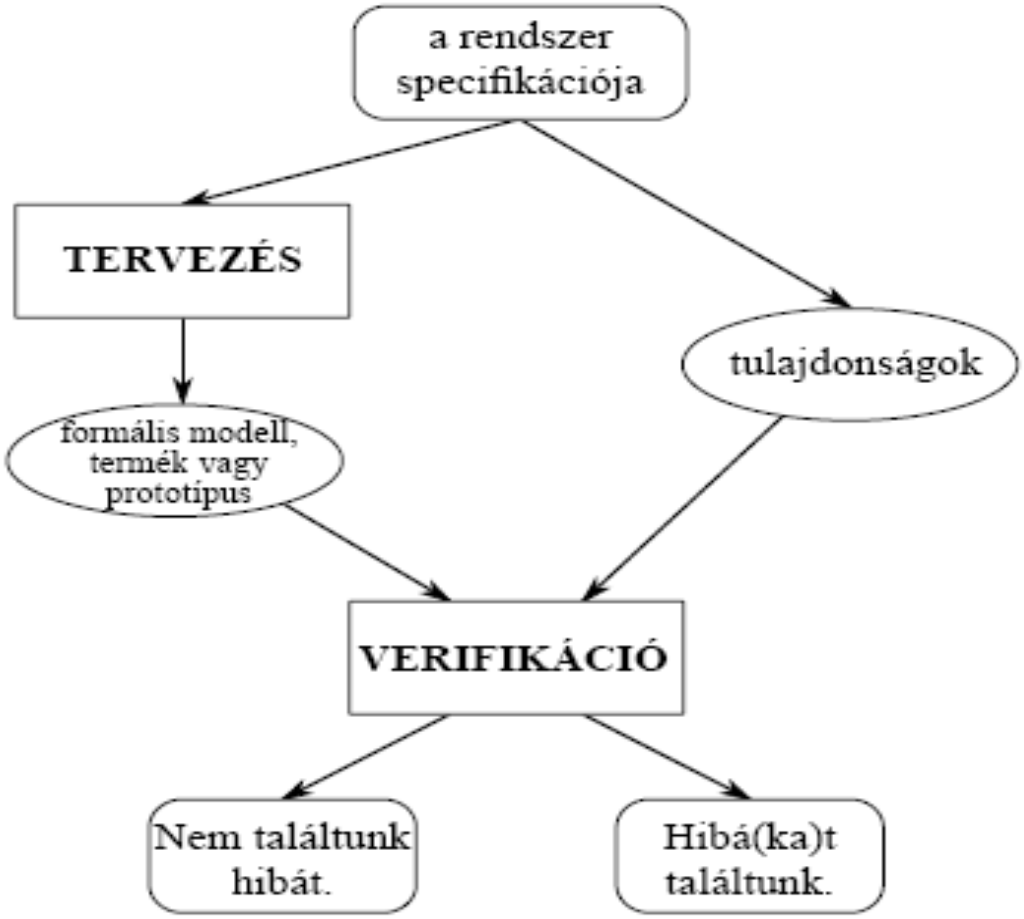
- informatikai rendszerek az élet számos területén megjelentek (pl. beágyazott rendszerek, speciális célú számítógépek megjelennek orvosi eszközökben, telefonban, stb., pl. közlekedés, ipar, vállalati irányítás, energiatermelés, üzleti és banki világ, oktatás területeken egyaránt),
- a rendszerek komplexitása nagymértékben nőtt (vezetékes és vezetés nélküli kapcsolat a komponensek között, egymásra hatás nehezen kiszámítható, hibalehetőségek száma növekszik);

Elvárás az informatikai rendszerek alkalmazásánál:

- Biztonságkritikusság
- Költségkritikusság

Informatikai rendszerek ellenőrzésének két megközelítése:

- Validáció: külső körülményeknek és a felhasználói elvárásokkal való megfelelést ellenőrzi (Jó rendszert építünk-e?)
- Verifikáció: azt ellenőrzi, hogy az adott fejlesztési fázis eredménye teljesíti-e a fejlesztési fázis kezdetén megfogalmazott feltételeket (Jól építjük-e a rendszert?)



1.1. ábra. A verifikáció általános menete

- elsődleges verifikáció: a rendszer formális modelljét ellenőrzi
- utólagos verifikáció*:* A prototípust vagy kész terméket vizsgálja

Verifikáció sohasem a rendszer abszolút helyességét igazolja, hanem mindig relatív, csak a specifikációnak való megfelelést képes biztosítani.

Szoftver verifikáció:

a verifikációnak a szoftverfejlesztés egész menetét végig kell kísérni;

technikák:

- Szakértői felülvizsgálat: a programkód futtatás nélküli, manuális ellenőrzése
- Szoftvertesztelés: tesztesetek generálása és végrehajtása, az eredmények összevetése a specifikációval;
- Formális módszerek (absztrakt statikus analízis, modell- ellenőrzés);

Hardver verifikációs technikák:

- strukturális elemzés
- emuláció (egyfajta tesztelés): az emulátort (általános célú hardver) úgy konfigurálnak, hogy viselkedése a tervezett hardver áramkört utánozza, majd ezt tesztelik);
- szimuláció: a vizsgálandó áramkör szoftveres modelljét (hardver leíró nyelv használatával) készítik el, majd ennek működését a szimulátor programmal vizsgálják;

A hibakeresési módszereken túl szükséges a kész hardver tesztelése is (gyártási hibák kiszűrése).

Modell-ellenőrzés

- 1980-as évek elejétől indult;
- E.M. Clarke, E. A. Emerson, J. Sifakis – 2007-ben Turing-díj (hatékony, ipari környezetben is alkalmazható verifikációs technika kidolgozásáért);
- matematikai elméleteken alapuló módszer, ami a módszer megbízhatóságának garanciája;
- a rendszermodell állapotainak szisztematikus bejárásával dönti el, hogy a rendszermodell a specifikációt teljesíti-e vagy nem;

Modell-ellenőrzés menete

modellezési fázis:

- a rendszer modelljének elkészítése (véges átmeneti rendszerek, modell leíró nyelv (Promela));
- a vizsgálandó tulajdonság valamely tulajdonságokat leíró nyelven való megadása (temporális logikák);

futtatási fázis:

a modell-ellenőrző program futtatása , aminek eredménye lehet: a tulajdonság teljesül vagy nem teljesül vagy a tár mérete kevés a vizsgált modell állapotainak bejárásához;

elemzési fázis*:*

ha a vizsgált tulajdonság nem teljesül, akkor a hiba okának felderítése;

ha a rendszermodell túl nagy:

- a rendszer modelljének további absztrakcióval való csökkentése;
- szimbolikus technika az állapottér reprezentálásánál ;
- valószínűségi ellenőrzés (lemond az állapottér teljes bejárásáról);
- korlátos modell-ellenőrzés (állapotokat Boole változókkal írja le; SAT megoldó program alkalmazása);

állapottér reprezentációja alapján lehet:

- explicit modell-ellenőrzés: állapotokat explicit módon tárol a memóriában (pl. Spin);
- szimbolikus modell-ellenőrzés: állapothalmaz szimbolikus reprezentálás (BDD) (pl. Uppaal);

Modell-ellenőrzés előnyei:

- általános (hardver, szoftver, beágyazott rendszerek) verifikációs módszer;
- parciális verifikációt támogatja;
- ritkán előforduló hibák felderítésére is alkalmas;
- diagnosztikai információt ad (pl. ellenpélda);
- gombnyomásra működő (push-button) technológia;
- könnyen integrálható létező hardver-szoftver fejlesztési ciklusba;
- szilárd matematikai elméleteken alapul;
- növekvő érdeklődés az ipar részéről;

Modell-ellenőrzés hátrányai:

- inkább vezérlés-intenzív rendszerekre alkalmazható;
- eldönthetetlenségi eredményekbe ütközhet;
- ellenőrzés a modellre történik, nem a valódi rendszerre;
- csak a specifikált tulajdonságokat vizsgálja;
- állapotszám robbanási probléma;
- bizonyos szakértelmet kíván;
- a modell-ellenőrző szoftver maga is lehet hibás;
- nem engedi meg az általánosítások igazolását (pl. parametrizált vagy tetszőlegesen sok komponensből álló rendszerek nem ellenőrizhetők);

Egyszerű átmeneti rendszerek

$\mathcal{A} = (S, T, \alpha, \beta)$, ahol

S az *állapotok* véges vagy végtelen halmaza,

T az *átmenetek* véges vagy végtelen halmaza,

$\alpha, \beta : T \rightarrow S$,

$\forall t \in T$-re $\alpha(t)$ a t kezdőpontja, $\beta(t)$ a t átmenet végpontja.

Gráf reprezentálás:

- címkézett irányított gráffal;
- csúcsok állapotokat reprezentálnak;
- élek átmeneteket reprezentálnak

$\forall t \in T$-re $\alpha(t)$ csúcsból él vezet $\beta(t)$ csúcsba;

$\mathcal{A}$ = (S, T, α, β) átmeneti rendszerben n>0 hosszú útnak nevezünk egy $t_1t_2\ldots t_n$ sorozatot, ha $\beta(t_i)=\alpha(t_{i+1})$ teljesül $\forall i\in\{1,\ldots,n-1\}$-re.

α és β függvények kiterjesztése útra:
$\alpha(t_1t_2\ldots t_n) = \alpha(t_1)$, $\beta(t_1t_2\ldots t_n) = \beta(t_n)$;
c üres út, ha hossza 0, $\alpha(c)= \beta(c)$;
ε_s üres út s-állapothoz;

Végtelen út olyan $t_1t_2\ldots t_n$ … végtelen sorozat, ahol $\beta(t_i)=\alpha(t_{i+1})$ teljesül $\forall i \geq 1$-re.

Út megadható az út által érintett állapotok sorozatával is (vagyis az utat alkotó átmenetek kezdőpontjai sorozatával, illetve véges útnál még ez kiegészül az utolsó átmenet végpontjával).

$c = t_1 t_2 \ldots t_n$ véges út, $c' = g_1 \ldots$ véges vagy végtelen út.

Ha $\beta(c) = \alpha(c')$, akkor definiálható a c és a c' utak konkatenációja, összefűzése (művelet jele $\cdot$, de gyakran nem írjuk ki):

$$c \cdot c' = t_1 t_2 \ldots t_n \, g_1 \ldots$$

$\alpha(\varepsilon_s) = \beta(\varepsilon_s) = s$, $c \cdot \varepsilon_s = c$, $\varepsilon_s \cdot c' = c'$, ha $\beta(c) = s$, $\alpha(c') = s$;

Ha $\beta(c) \neq \alpha(c')$, akkor a c és a c' utak összefűzését nem definiáljuk.

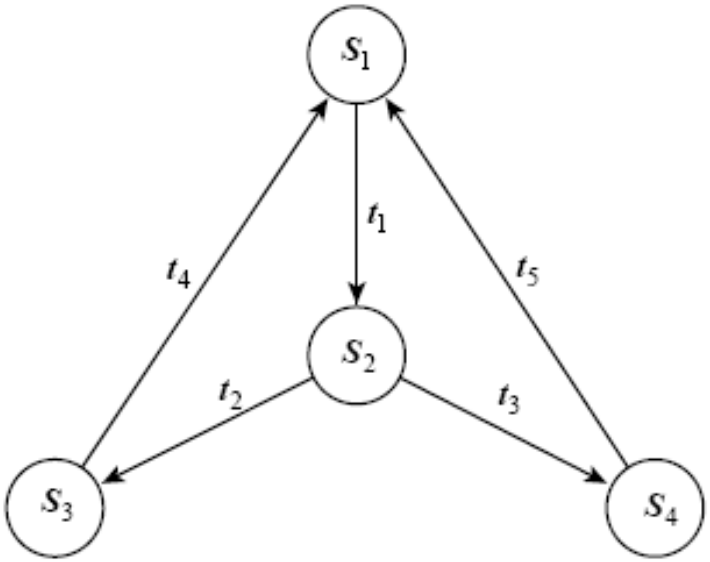
példa: italautomata egyszerű átmeneti rendszere

- rögzített összeg dobható be;
- két gomb van, egyikkel kávé (K), másikkal tea (T) kérhető;
- választott italt kiadja, majd ismét pénz bedobásra vár;

állapotok: s_1: pénz bedobásra vár;
s_2: K vagy T gomb megnyomására vár;
s_3: K gombot nyomták meg, kávét kell kiadni;
s_4: T gombot nyomták meg, teát kell kiadni;

átmenetek: t_1: pénz bedobása;
t_2: K gomb megnyomása;
t_3: T gombot megnyomása;
t_4: kávét kiad;
t_5: teát kiad;

példa: italautomata működését formálisan megadó átmeneti rendszer gráf reprezentációja



példa útra: $c = t_4 t_1 t_3 t_5 t_1$, hossza 5, $\alpha(c) = s_3$, $\beta(c) = s_2$;

Címkézett átmeneti rendszerek

A egy véges vagy végtelen nem üres halmaz (ábécé);

$\mathcal{A} = (S, T, \alpha, \beta, \lambda)$ ötös *A* feletti címkézett átmeneti rendszer, ha (S, T, α, β) egyszerű átmeneti rendszer,

$\lambda: T \to A$ leképezés .

$\forall t \in T$-re $\lambda(t)$ a *t* átmenet címkéje;

A elemeit akciónak vagy eseménynek nevezzük;

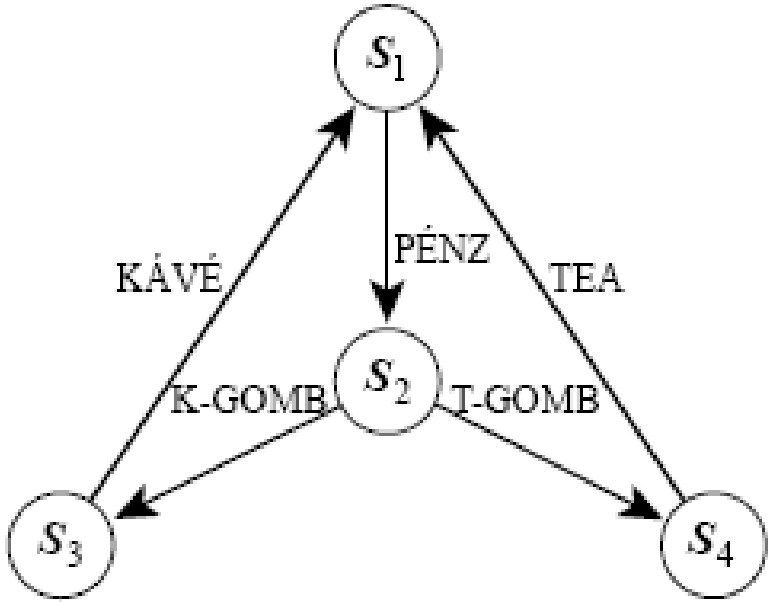
Feltesszük, hogy nem lehet két olyan átmenet, melyeknek kezdőpontja, végpontja és címkéje is azonos, vagyis ez a három adat egyértelműen meghatározza az átmeneteket.

jelölés: ha $\alpha(t)=s$, $\beta(t)=s'$, $\lambda(t)=a$, akkor

$$t : s \mapsto a \rightarrow s'$$

$c= t_1 t_2 \ldots$ véges vagy végtelen út az $\mathcal{A} = (S, T, \alpha, \beta, \lambda)$ címkézett átmeneti rendszerben, ekkor a *c* út nyoma : *trace* $(c)=\lambda(t_1)\ \lambda(t_2)\ldots$

példa: italautomata formális modellje címkézett átmeneti rendszerrel



A={PÉNZ, K-GOMB, T-GOMB, KÁVÉ, TEA}

$\lambda(t_1)$=PÉNZ, $\lambda(t_2)$=K-GOMB,

$\lambda(t_3)$= T-GOMB, $\lambda(t_4)$= KÁVÉ,

$\lambda(t_5)$= TEA

példa út lenyomatra:

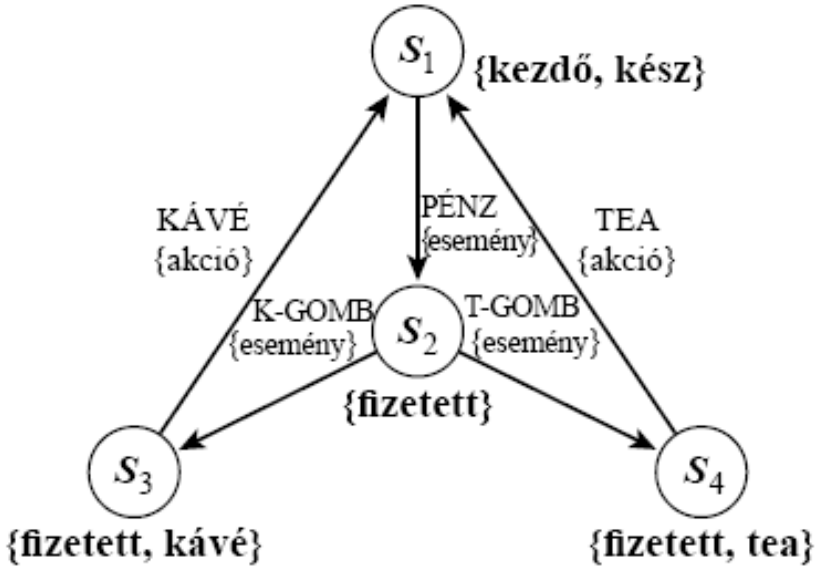
trace($t_4t_1t_3t_5t_1$)=KÁVÉ PÉNZ T-GOMB TEA PÉNZ

Paraméteres átmeneti rendszerek

$X=\{x_1,\ldots, x_n\}$, $Y=\{y_1,\ldots, y_m\}$ véges halmazok;
X elemei állapotparaméterek,
Y elemei átmenetparaméterek;

$\mathcal{A} = (S, T, \alpha, \beta, S_{x_1},\ldots, S_{x_n}, T_{y_1}, \ldots, T_{y_m})$ rendszert
(X, Y)-paraméterezett átmeneti rendszernek nevezzük,
ha (S, T, α, β) egyszerű átmeneti rendszer,
$S_{x_i} \subseteq S$, $T_{y_j} \subseteq T$ minden $1 \leq i \leq n$, $1 \leq j \leq m$.

példa: italautomata formális modellje paraméteres átmeneti rendszerrel



X={kezdő, kész, fizetett, kávé, tea},

Y={esemény, akció},

$S_{\text{kezdő}}=\{s_1\}$, $S_{\text{kész}}=\{s_1\}$,

$S_{\text{fizetett}}=\{s_2, s_3, s_4\}$, $S_{\text{kávé}}=\{s_3\}$,

$S_{\text{tea}}=\{s_4\}$,

$T_{\text{esemény}}=\{t_1, t_2, t_3\}$,

$T_{\text{akció}}=\{t_4, t_5\}$

megjegyzés:

Minden egyszerű átmeneti rendszer tekinthető olyan címkézett átmeneti rendszernek, ahol minden átmenet önmagával van címkézve.

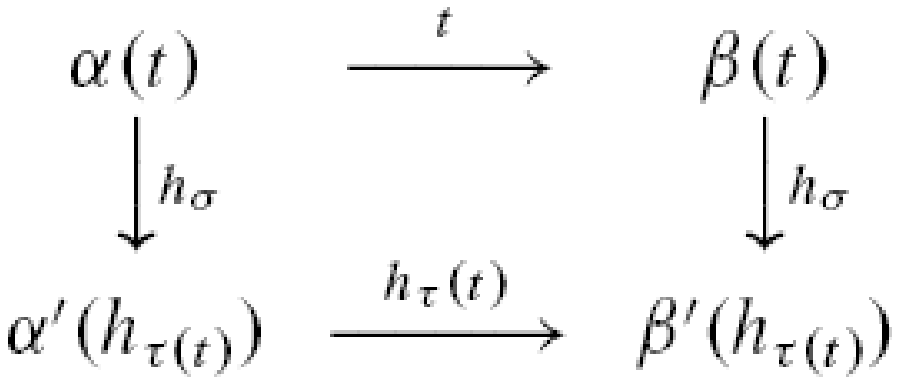
Minden A-feletti címkézett átmeneti rendszer tekinthető olyan $(\emptyset,A)$-paraméterezett átmeneti rendszernek, melyben az átmenetek a címkéjüknek megfelelő paraméterrel vannak ellátva.

Átmeneti rendszerek homomorfizmusai

$\mathcal{A} = (S, T, \alpha, \beta)$ és $\mathcal{A}' = (S', T', \alpha', \beta')$ egyszerű átmeneti rendszerek;

$\mathcal{A}$-ból $\mathcal{A}'$-be ható homomorfizmus olyan $h = (h_\sigma, h_\tau)$ leképezés pár, melyre: $h_\sigma : S \to S'$, $h_\tau : T \to T'$,

$\forall t \in$ T-re : $\alpha'(h_\tau(t)) = h_\sigma(\alpha(t))$ és $\beta'(h_\tau(t)) = h_\sigma(\beta(t))$.



$\mathcal{A}$ = (S, T, α, β, λ) és $\mathcal{A}'$ = (S', T', α', β', λ') azonos A címkehalmaz feletti címkézett átmeneti rendszerek;

$\mathcal{A}$-ból $\mathcal{A}'$-be ható címkézett átmeneti rendszerek között homomorfizmus olyan $h = (h_\sigma, h_\tau)$ leképezés pár, mely (S, T, α, β)-ból (S', T', α', β')-be ható homomorfizmus és $\forall\ t \in T$-re $\lambda'(h_\tau(t)) = \lambda(t)$.

Egyszerű vagy címkézett átmeneti rendszerek közötti izomorfizmus olyan homomorfizmus, mely bijektív.

Nem teszünk különbséget izomorf átmeneti rendszerek között, mivel egymástól csak az állapotok és átmenetek elnevezésében térnek el.

$\mathcal{A} = (S, T, \alpha, \beta, S_{x_1}, \ldots, S_{x_n}, T_{y_1}, \ldots, T_{y_m})$ és $\mathcal{A}' = (S', T', \alpha', \beta', S'_{x_1}, \ldots, S'_{x_n}, T'_{y_1}, \ldots, T'_{y_m})$ azonos (X, Y) feletti paraméterezett átmeneti rendszerek;

$\mathcal{A}$-ból $\mathcal{A}'$-be ható paraméterezett átmeneti rendszerek közötti homomorfizmus olyan $h = (h_\sigma, h_\tau)$ leképezés pár, mely (S, T, α, β)-ból $(S', T', \alpha', \beta')$-ba ható homomorfizmus, továbbá teljesül

$\forall s \in S, \forall x \in X : (s \in S_x \Leftrightarrow h_\sigma(s) \in S'_x)$

$\forall t \in T, \forall y \in Y : (t \in T_y \Leftrightarrow h_\tau(t) \in T'_y)$.

$\mathcal{A}$ *és* $\mathcal{A}'$ (X, Y) feletti paraméterezett átmeneti rendszerek izomorfak, ha van közöttük h homomorf leképezés, mely bijektív.

Példák formális modellekre

Boole változó

események:

- üres esemény (**e**)
- értékadás a változónak (**b := true**, **b := false**)
- a változók értékének kiolvasása (**true!**, **false!**)

rendszer átmenetei:

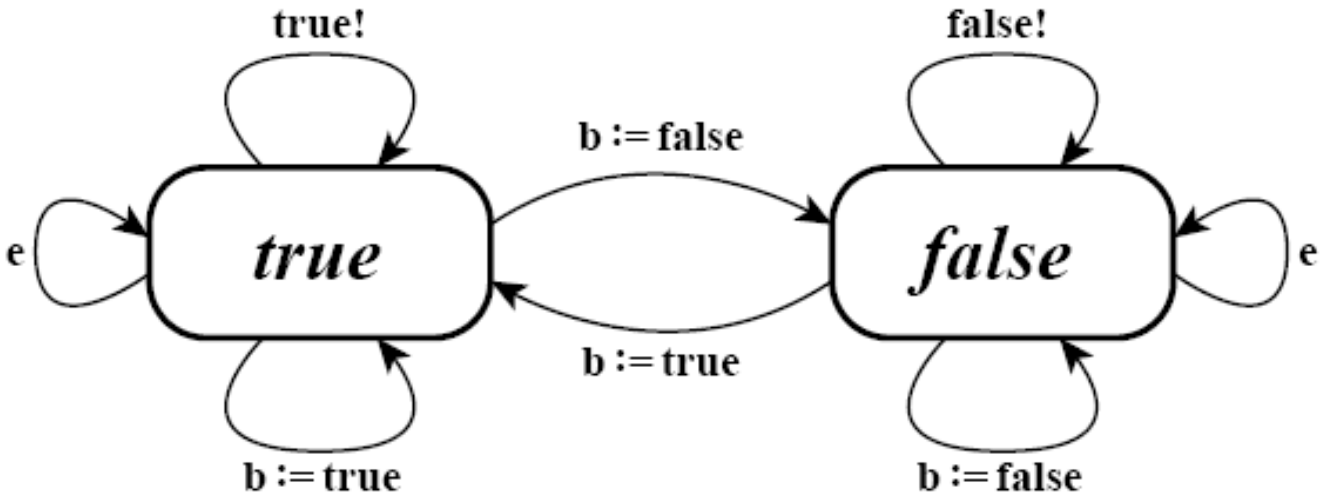
t_1 : true $\mapsto$ **b:=true** $\rightarrow$ true; t_5: true $\mapsto$ **true!** $\rightarrow$true

t_2 : true $\mapsto$ **b:=false** $\rightarrow$ false; t_6 : false $\mapsto$ **false!** $\rightarrow$false

t_3 : false $\mapsto$ **b:=true** $\rightarrow$ true; t_7 : true $\mapsto$ **e** $\rightarrow$ true

t_4 : false $\mapsto$ **b:=false** $\rightarrow$ false; t_8 : false $\mapsto$ **e** $\rightarrow$ false

példa: Boole változó működését formálisan megadó átmeneti rendszer gráf reprezentációja

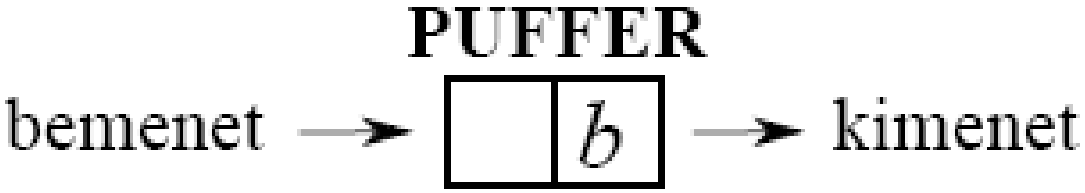


Korlátos pufferek

max. 2 szimbólum (a, b) tárolása

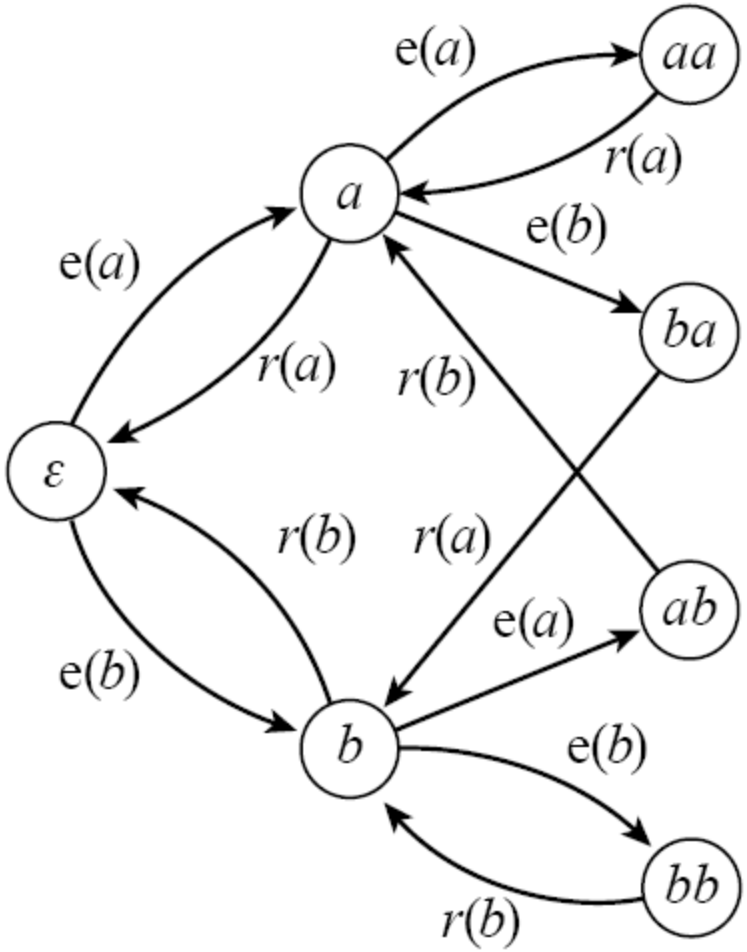
események:

- új betű behelyezése (**enter (a)**, **enter(b)**)
- egy betű kivétele (**remove(a)**, **remove(b)**)



átmenetek:

$$\varepsilon \mapsto \textbf{enter}(\mathrm{a}) \to a, \qquad a \mapsto \textbf{remove}(\mathrm{a}) \to \varepsilon,$$
$$\varepsilon \mapsto \textbf{enter}(\mathrm{b}) \to b, \qquad b \mapsto \textbf{remove}(\mathrm{b}) \to \varepsilon,$$
$$a \mapsto \textbf{enter}(\mathrm{a}) \to aa, \qquad aa \mapsto \textbf{remove}(\mathrm{a}) \to a,$$
$$a \mapsto \textbf{enter}(\mathrm{b}) \to ba, \qquad ab \mapsto \textbf{remove}(\mathrm{b}) \to a,$$
$$b \mapsto \textbf{enter}(\mathrm{a}) \to ab, \qquad ba \mapsto \textbf{remove}(\mathrm{a}) \to b,$$
$$b \mapsto \textbf{enter}(\mathrm{b}) \to bb, \qquad bb \mapsto \textbf{remove}(\mathrm{b}) \to b.$$



2.8. ábra. A puffer címkézett átmeneti rendszer modellje,
e(x) *az* **enter(x)**, **r(x)** *a* **remove(x)** *címke rövidítése*

Szekvenciális program

```
while true do
    1: if not b then
       begin
           2: b:=true;
           3: proc;
           4: b:=false;
       end
```

szekvenciális programhoz megfeleltethető rendszer *átmenetei*:

t_1 : $1 \mapsto$ b=true? $\rightarrow 1$;

t_2 : $1 \mapsto$ b=false? $\rightarrow 2$;

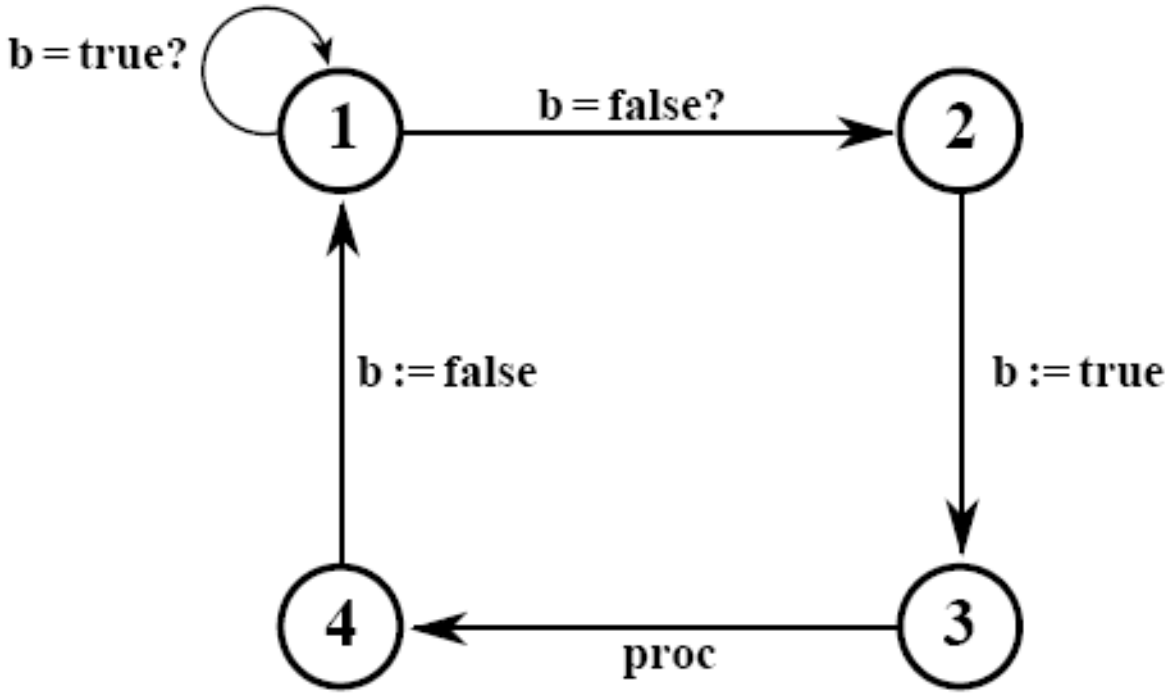
t_3 : $2 \mapsto$ b:=true $\rightarrow 3$;

t_4 : $3 \mapsto$ proc $\rightarrow 4$;

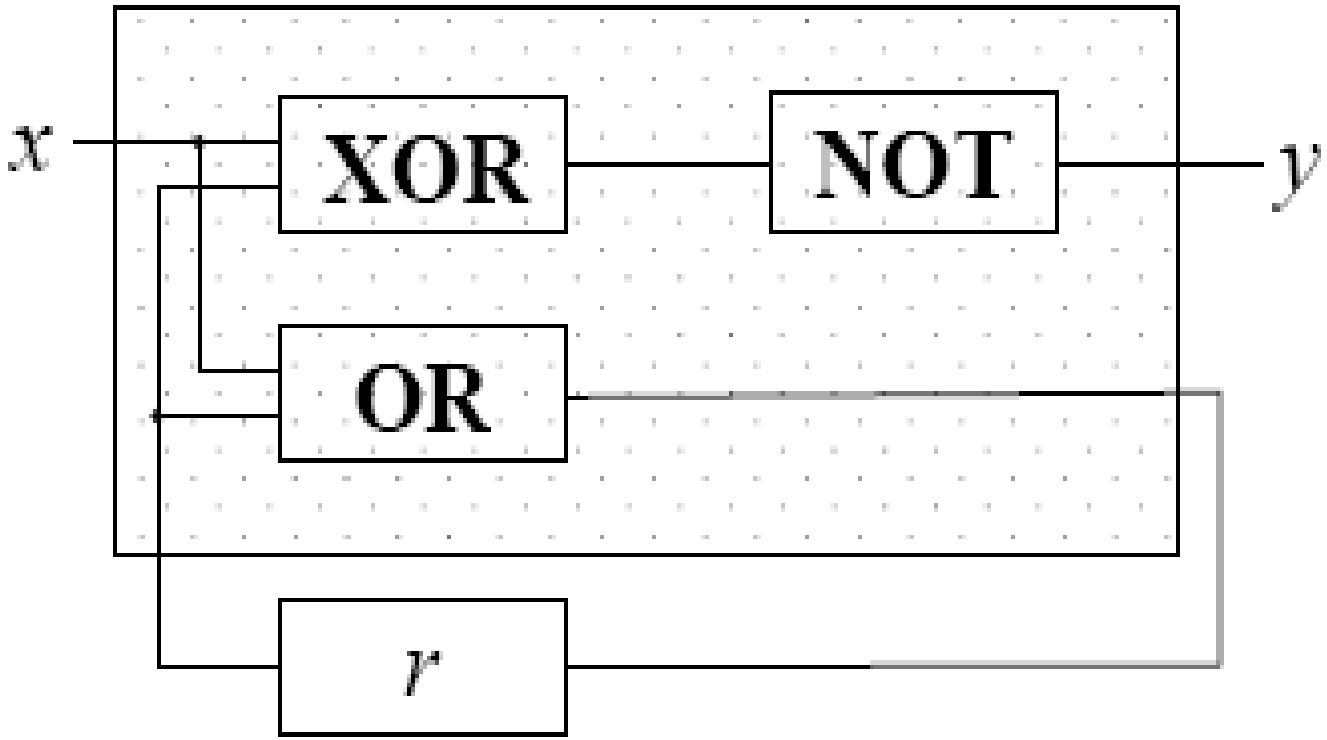
t_5 : $4 \mapsto$ b:=false $\rightarrow 1$;

kezdőállapot: 1

példa: szekvenciális programból származtatott átmeneti rendszer gráf reprezentációja

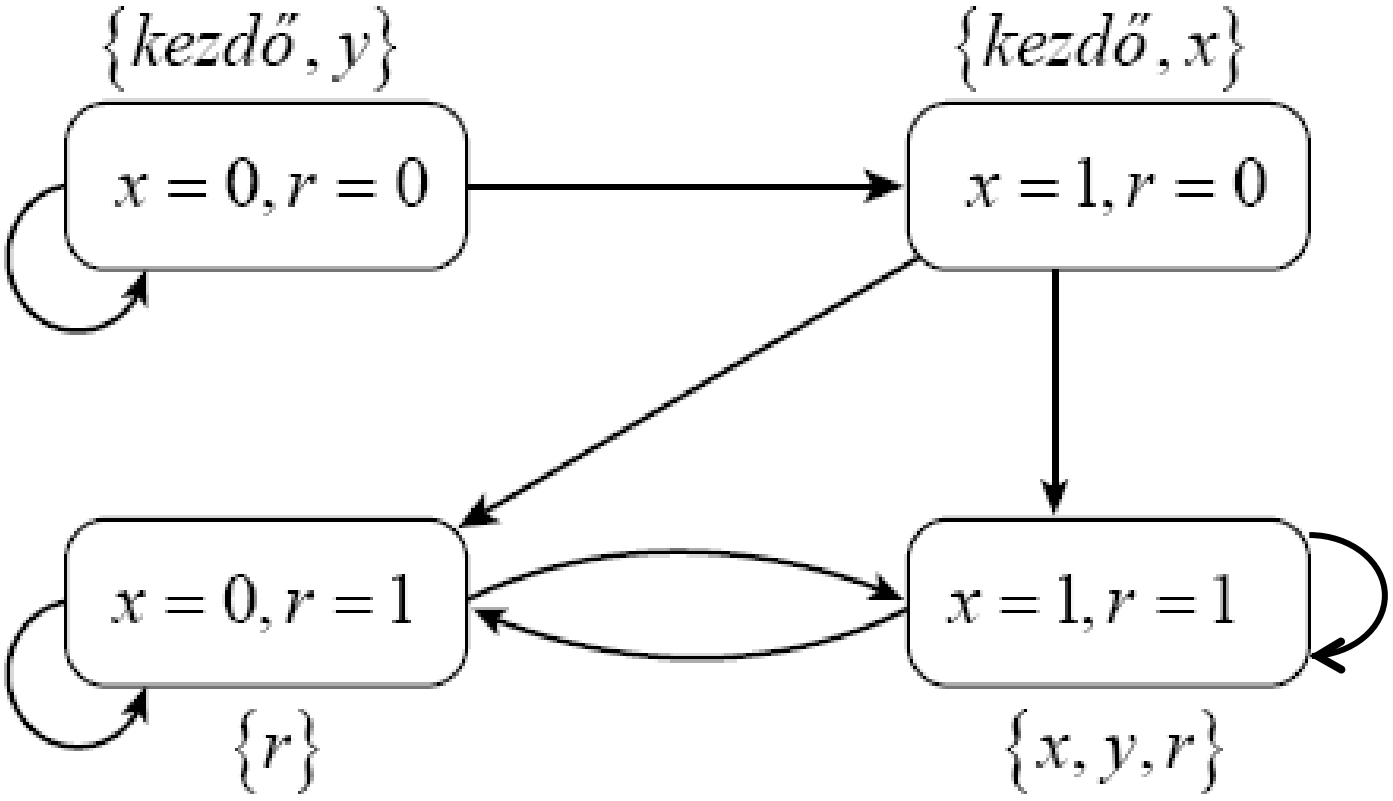


Szekvenciális áramkörök



X= {kezdő, x, y, r}

$S_{kezdő}$={[x=0, r=0], [x=1, r=0]}, S_x = {[x=1, r=1], [x=1, r=0]}, S_y = {[x=0, r=0], [x=1, r=1]}, S_r={[x=0, r=1], [x=1, r=1]}



Peterson algoritmus

- processzusok: P_0 és P_1;
- közös erőforrás kizárólagos kezelése (kritikus szekció);
- globális változók: *d0*, *d1* Boole változók, *turn* egész értékű változó;
- P_0 a *d0*-t igazra állítja mielőtt a kritikus szekcióba lép;
- P_1 a *d1*-t igazra állítja mielőtt a kritikus szekcióba lép;
- *turn* értékének megfelelő processzus kerülhet a kritikus szekcióba, ha *d0* és *d1* értéke is igaz – biztosítja, hogy a processzusok fair módon férjenek a kritikus szekcióhoz;

P_0 az alábbi kódot futtatja:

```
while true do
begin
   1: {non-critical section}
   2: d0:=true;
   3: turn:=1;
   4: wait_until(d1=false or turn=0);
   5: {critical section}
   6: d0:=false;
end
```

P_1 kódja: hasonló P_0 kódjához csupán a 0-kat és 1-eket fel kell cserélni;

P_0 átmenetei:

$t_1 : 1 \mapsto \textbf{ncs} \rightarrow 2,$

$t_2 : 2 \mapsto \textbf{d0:=true} \rightarrow 3,$

$t_3 : 3 \mapsto \textbf{turn:=1} \rightarrow 4,$

$t_4 : 4 \mapsto \textbf{d1=false?} \rightarrow 5,$

$t_5 : 4 \mapsto \textbf{turn=0?} \rightarrow 5,$

$t_6 : 5 \mapsto \textbf{cs} \rightarrow 6,$

$t_7 : 6 \mapsto \textbf{d0:=false} \rightarrow 1.$

P_1 átmenetei:

$t'_1 : 1 \mapsto \textbf{ncs} \rightarrow 2,$

$t'_2 : 2 \mapsto \textbf{d1:=true} \rightarrow 3,$

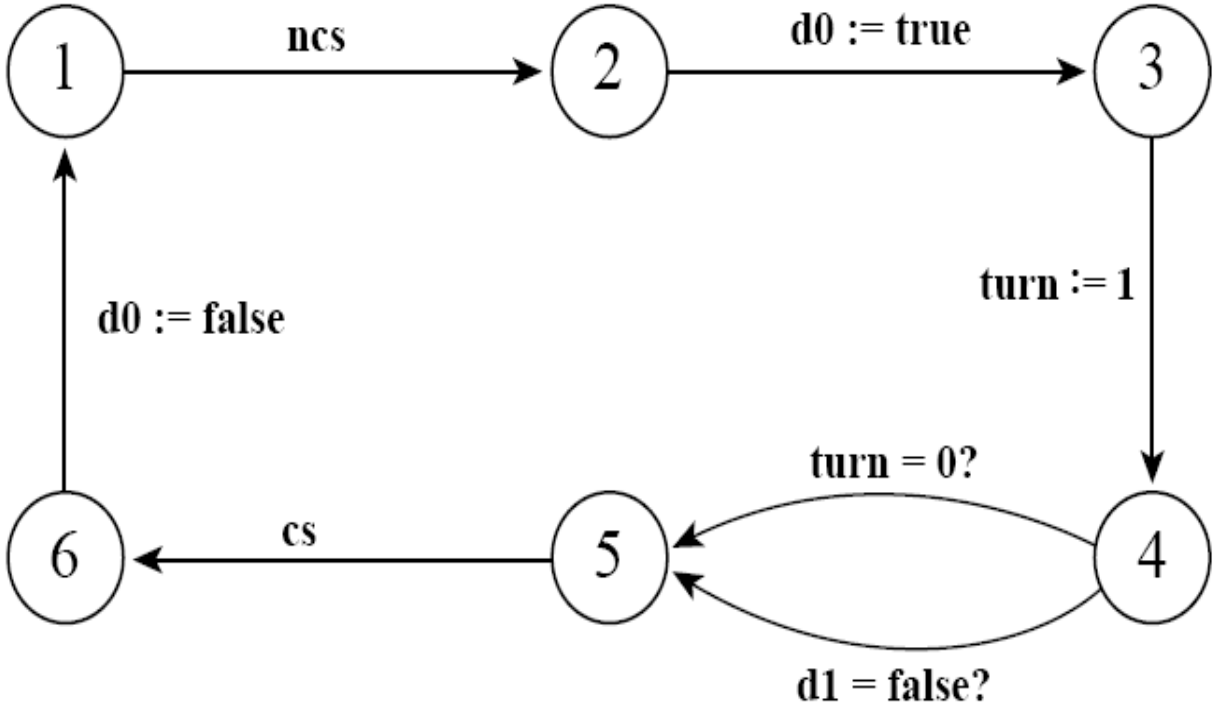
$t'_3 : 3 \mapsto \textbf{turn:=0} \rightarrow 4,$

$t'_4 : 4 \mapsto \textbf{d0=false?} \rightarrow 5,$

$t'_5 : 4 \mapsto \textbf{turn=1?} \rightarrow 5,$

$t'_6 : 5 \mapsto \textbf{cs} \rightarrow 6,$

$t'_7 : 6 \mapsto \textbf{d1:=false} \rightarrow 1.$



2.10. ábra. A Petterson algoritmus P_0 processzusát modellező címkézett átmeneti rendszer

[Animáció]

Átmeneti rendszerek szabad szorzata

(komponensek egymástól független működnek)

$\mathcal{A}_i = (S_i, T_i, \alpha_i, \beta_i)$, $i \in \{1,\dots,n\}$, átmeneti rendszerek;

$\mathcal{A}_i$-k szabad szorzata $\mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_n = (S, T, \alpha, \beta)$ átmeneti rendszer, ahol

$S = S_1 \times S_2 \times \dots \times S_n$,

$T = T_1 \times T_2 \times \dots \times T_n$,

$\alpha((t_1,\dots,t_n)) = (\alpha_1(t_1), \dots, \alpha_n(t_n))$,

$\beta((t_1,\dots,t_n)) = (\beta_1(t_1), \dots, \beta_n(t_n))$.

$\mathcal{A}_i = (S_i, T_i, \alpha_i, \beta_i, \lambda_i)$ A_i feletti címkézett átmeneti rendszerek, $i \in \{1,\ldots,n\}$-re;

$\mathcal{A}_i$ címkézett átmeneti rendszerek szabad szorzata

$\mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n = (S, T, \alpha, \beta, \lambda)$ címkézett átmeneti rendszer $A_1 \times \ldots \times A_n$ felett, ahol S, T, α, β, mint az egyszerű átmeneti rendszerek szabad szorzatánál, $\lambda : T \rightarrow A_1 \times \ldots \times A_n$ címkefüggvényre teljesül $\lambda((t_1,\ldots,t_n)) = \langle(\lambda_1(t_1), \ldots, \lambda_n(t_n)\rangle \; \forall \; (t_1,\ldots,t_n) \in T$-re.

megjegyzés*:*

A szabad szorzat állapotait globális állapotoknak, átmeneteit globális átmeneteknek nevezzük. Címkézett átmenetei rendszerek esetén a t globális átmenet $\lambda(t)$ címkéjét globális akciónak nevezik.

Szinkron módon működő rendszerek:
minden komponens minden lépésben (globális óraütésre) egy átmenetet végrehajt;

Aszinkron módon működő rendszerek:
megengedett, hogy egy-egy komponens tetszőleges ideig maradhasson egy állapotban;

Kevert, szinkron-aszinkron rendszerek:
tétlen várakozást nem mindig, hanem csak bizonyos állapotokban tehetnek;

Aszinkron rendszerek modellezhetők szinkron módon *üres események* vagy *üres akciók* (tétlen várakozás, *e* címkéjű átmenetek) alkalmazásával.

példa*:* Peterson algoritmus két processzusának egyetlen processzorral való végrehajtásánál minden pillanatban az egyik processzus inaktív.

Kikötve, hogy az átmeneti rendszerben mindig pontosan egy komponens hajt végre nem *e* átmenetet, a többi komponens pedig *e* átmenetet, akkor az *e* átmenetekkel kibővített rendszer szinkron működésével az aszinkron működést modellezi.

Továbbiakban a fentiek miatt csak szinkron működést tárgyalunk.

Átmeneti rendszerek szinkronizált szorzata

(kommunikációs megszorítások fejezhetők ki vele)

$\mathcal{A}_i$ az A_i feletti címkézett átmeneti rendszer minden $i \in \{1,\dots,n\}$-re és $I \subseteq A_1 \times \dots \times A_n$ szinkronizációs megszorítás ;

$\mathcal{A}_1, \dots, \mathcal{A}_n$ átmeneti rendszerek I szerinti szinkronizált szorzata az $\mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_n$ szabad szorzat azon rész átmeneti rendszere, mely pontosan azokat a $t=(t_1,\dots,t_n)$ globális átmeneteket tartalmazza, melyekre $\langle \lambda_1(t_1), \dots, \lambda_n(t_n) \rangle \in I$ teljesül.

szinkronizációs vektorok: a szinkronizációs megszorítás elemei;

példa:

A korábbi *P* szekvenciális programot és *B* Boole változót reprezentáló címkézett átmeneti rendszerből álló $P \times B$ szabadszorzat megengedett globális akciói:

$$\langle \text{ b:= true , b:=true } \rangle$$

$$\langle \text{ b:= false , b:=false } \rangle$$

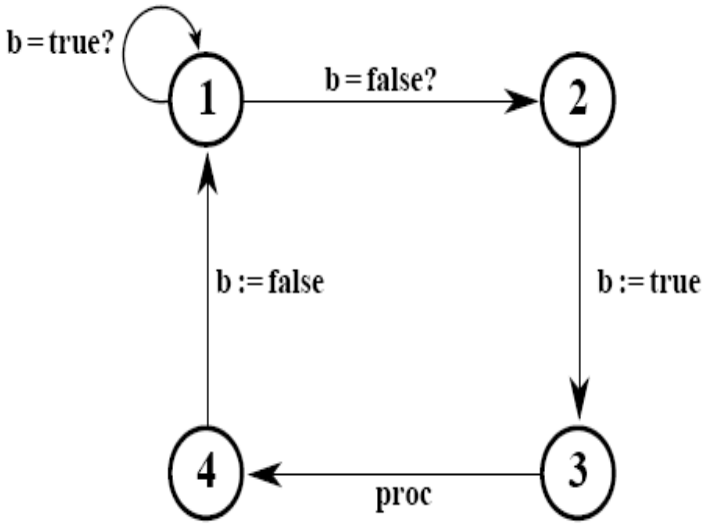
$$\langle \text{ b:= true? , b:=true! } \rangle$$

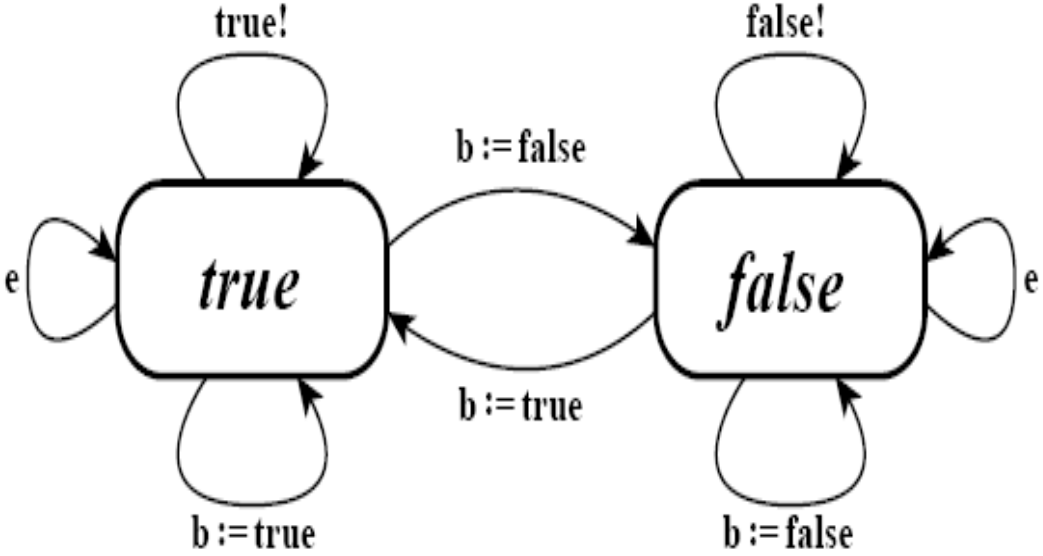
$$\langle \text{ b:= false? , b:=false! } \rangle$$

$$\langle \text{ proc , e } \rangle$$

Ezek a globális akciók a rendszer szinkronizációs vektorai.

$$
\begin{aligned}
(1, \mathbf{true}) &\mapsto \langle\, \mathbf{b = true?}, \mathbf{true!} \,\rangle \rightarrow (1, \mathbf{true}),\\
(2, \mathbf{true}) &\mapsto \langle\, \mathbf{b := true}, \mathbf{b := true} \,\rangle \rightarrow (3, \mathbf{true}),\\
(3, \mathbf{true}) &\mapsto \langle\, \mathbf{proc}, \mathbf{e} \,\rangle \rightarrow (4, \mathbf{true}),\\
(4, \mathbf{true}) &\mapsto \langle\, \mathbf{b := false}, \mathbf{b := false} \,\rangle \rightarrow (1, \mathbf{false}),\\
(1, \mathbf{false}) &\mapsto \langle\, \mathbf{b = false?}, \mathbf{false!} \,\rangle \rightarrow (2, \mathbf{false}),\\
(2, \mathbf{false}) &\mapsto \langle\, \mathbf{b := true}, \mathbf{b := true} \,\rangle \rightarrow (3, \mathbf{true}),\\
(3, \mathbf{false}) &\mapsto \langle\, \mathbf{proc}, \mathbf{e} \,\rangle \rightarrow (4, \mathbf{false}),\\
(4, \mathbf{false}) &\mapsto \langle\, \mathbf{b := false}, \mathbf{b := false} \,\rangle \rightarrow (1, \mathbf{false}).
\end{aligned}
$$





Ha a kezdő globális állapot (**1, false**), akkor az alábbi átmenetek lesznek (a program determinisztikus):

$$
\begin{aligned}
(1, \mathbf{false}) &\mapsto \langle\, \mathbf{b} = \mathbf{false}?, \mathbf{false}! \,\rangle \rightarrow (2, \mathbf{false}),\\
(2, \mathbf{false}) &\mapsto \langle\, \mathbf{b} := \mathbf{true}, \mathbf{b} := \mathbf{true} \,\rangle \rightarrow (3, \mathbf{true}),\\
(3, \mathbf{true}) &\mapsto \langle\, \mathbf{proc}, \mathbf{e} \,\rangle \rightarrow (4, \mathbf{true}),\\
(4, \mathbf{true}) &\mapsto \langle\, \mathbf{b} := \mathbf{false}, \mathbf{b} := \mathbf{false} \,\rangle \rightarrow (1, \mathbf{false}).
\end{aligned}
$$

Ha a kezdő globális állapot (**1, true**), akkor ez lesz a rendszer egyetlen globális állapota, illetve egyetlen átmenete a rendszernek az

$$(1, \mathbf{true}) \mapsto \langle\, \mathbf{b} = \mathbf{true}?, \mathbf{true}! \,\rangle \rightarrow (1, \mathbf{true})$$

Paraméteres átmeneti rendszerek szinkronizált szorzata

$\mathcal{A}_i = (S_i, T_i, \alpha_i, \beta_i, \lambda_i)$ A_i feletti címkézett átmeneti rendszer, (X_i, Y_i)-paraméteres átmeneti rendszer $\forall i \in \{1,\dots,n\}$-re,

$I \subseteq A_1 \times \dots \times A_n$ szinkronizációs megszorítás;

Legyen $\mathcal{B} = (S, T, \alpha, \beta, \lambda)$ az $\mathcal{A}_i$-k I szerinti szinkronizált szorzata vagyis $\mathcal{B}$ I feletti címkézett átmeneti rendszer.

Legyen $X = \{(i, x) \mid 1 \leq i \leq n, x \in X_i\}$, $Y = \{(i, y) \mid 1 \leq i \leq n, y \in Y_i\}$,

$S_{(i,x)} = \{(s_1,\dots, s_i,\dots, s_n) \in S \mid s_i \in (S_i)_x\}$,

$T_{(i,y)} = \{(t_1,\dots, t_i,\dots, t_n) \in T \mid t_i \in (T_i)_y\}$.

$\mathcal{B}$-t kiegészítve a $S_{(i,x)}$, $T_{(i,y)}$ halmazokkal $\forall (i, x) \in X$, $(i, y) \in Y$ esetén, az előálló (X, Y)-paraméteres átmeneti rendszer az $\mathcal{A}_i$-k *I* szerinti szinkronizált szorzata.

Paraméteres átmeneti rendszerek szinkronizált szorzata megőrzi a részrendszerek paraméterezését.

pl. ha az $\mathscr{A}_i$ részrendszer $s \in S_i$, $x \in X_i$, $s \in (S_i)_x$, akkor $(s_1, \ldots, s_i, \ldots, s_n) \in S_{(i, x)}$, ahol $s_i = s$.

példa: alternáló bit protokoll (ABP, **A**lternating **B**it **P**rotocol)

feladata:

egy *S adó* (sender) és egy *R vevő* (receiver) között üzenetek megbízható továbbítása olyan csatornán keresztül, melyben az üzenetek megsérülhetnek, elveszhetnek.

feltesszük:

- egy időben egyirányú a csatorna (eredeti modellben kétirányú (full-duplex))
- üzenetek továbbítása azonnal történik (nem pufferen keresztül küldi)
- feltesszük, hogy minden hibás üzenetküldés a nem megfelelő sorszám bit észlelésével érzékelhető (nem különböztetjük meg a hibás ellenőrző bittel vett üzenetet és egyéb átviteli hibát).

megvalósítás:

S minden üzenethez egybites sorszámot csatol. Ismétli az üzenet küldését amíg a küldött üzenet sorszámát tartalmazó nyugta nem érkezik az *R*-től. Ekkor a sorszámbitet átfordítja és ezzel a sorszámmal küldi a következő üzenetet.

R üzenet hibás átvétele után ismétli az üzenet nyugtázását a fogadott üzenet sorszámával amíg egy ellentétes sorszámbittel ellátott üzenet érkezik hozzá. Ezután az új üzenetről kezd nyugtát küldeni egészen a következő üzenet sikeres vételéig.

S adó akciói:

em1 üzenet küldése 1-es bittel,
em0 üzenet küldése 0-ás bittel,
ra1 nyugta fogadása 1-es bittel,
ra0 nyugta fogadása 0-ás bittel.

S adó állapotai:

$send_0$, $send_1$, $wait_0$, $wait_1$, $resend_0$, $resend_1$

R vevő akciói:

rm1 üzenet fogadása 1-es bittel,
rm0 üzenet fogadása 0-ás bittel,
ea1 nyugta küldése 1-es bittel,
ea0 nyugta küldése 0-ás bittel.

R vevő állapotai:

$send_0$, $send_1$, $wait_0$, $wait_1$, $resend_0$, $resend_1$

Az adót modellező átmeneti rendszer:

$$
\begin{array}{lcl}
t_1 : & \mathbf{send}_0 & \mapsto \mathbf{em0} \rightarrow \mathbf{wait}_0, \\
t_2 : & \mathbf{send}_1 & \mapsto \mathbf{em1} \rightarrow \mathbf{wait}_1, \\
t_3 : & \mathbf{wait}_0 & \mapsto \mathbf{ra0} \rightarrow \mathbf{send}_1, \\
t_4 : & \mathbf{wait}_0 & \mapsto \mathbf{ra1} \rightarrow \mathbf{resend}_0, \\
t_5 : & \mathbf{wait}_1 & \mapsto \mathbf{ra1} \rightarrow \mathbf{send}_0, \\
t_6 : & \mathbf{wait}_1 & \mapsto \mathbf{ra0} \rightarrow \mathbf{resend}_1, \\
t_7 : & \mathbf{resend}_0 & \mapsto \mathbf{em0} \rightarrow \mathbf{wait}_0, \\
t_8 : & \mathbf{resend}_1 & \mapsto \mathbf{em1} \rightarrow \mathbf{wait}_1.
\end{array}
$$

$X = \{initial\}$, $Y = \{emission, re\text{-}emission\}$

$S_{initial} = \{send_0, send_1\}$

$T_{emission} = \{t_1, t_2\}$, első alkalommal történő üzenetküldés

$T_{re\text{-}emission} = \{t_7, t_8\}$, üzenetküldés megismétlése

Vevőt modellező átmeneti rendszer:

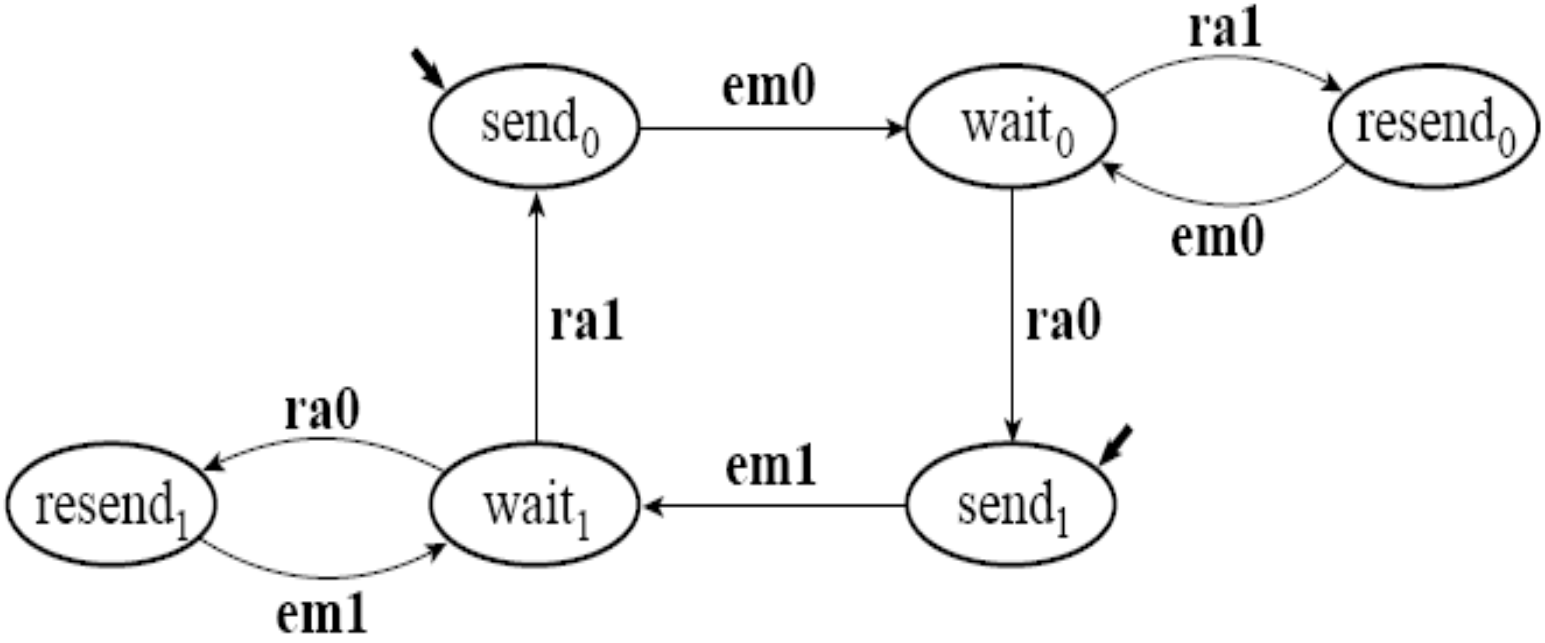
$$
\begin{array}{lrlll}
t_1' & : & \mathbf{wait}_0 & \mapsto \mathbf{rm0} \rightarrow & \mathbf{send}_0, \\
t_2' & : & \mathbf{wait}_0 & \mapsto \mathbf{rm1} \rightarrow & \mathbf{resend}_1, \\
t_3' & : & \mathbf{wait}_1 & \mapsto \mathbf{rm1} \rightarrow & \mathbf{send}_1, \\
t_4' & : & \mathbf{wait}_1 & \mapsto \mathbf{rm0} \rightarrow & \mathbf{resend}_0, \\
t_5' & : & \mathbf{send}_0 & \mapsto \mathbf{ea0} \rightarrow & \mathbf{wait}_1, \\
t_6' & : & \mathbf{send}_1 & \mapsto \mathbf{ea1} \rightarrow & \mathbf{wait}_0, \\
t_7' & : & \mathbf{resend}_0 & \mapsto \mathbf{ea0} \rightarrow & \mathbf{wait}_1, \\
t_8' & : & \mathbf{resend}_1 & \mapsto \mathbf{ea1} \rightarrow & \mathbf{wait}_0.
\end{array}
$$

$X = \{initial\}$, $Y = \{well\_received, ill\_received\}$

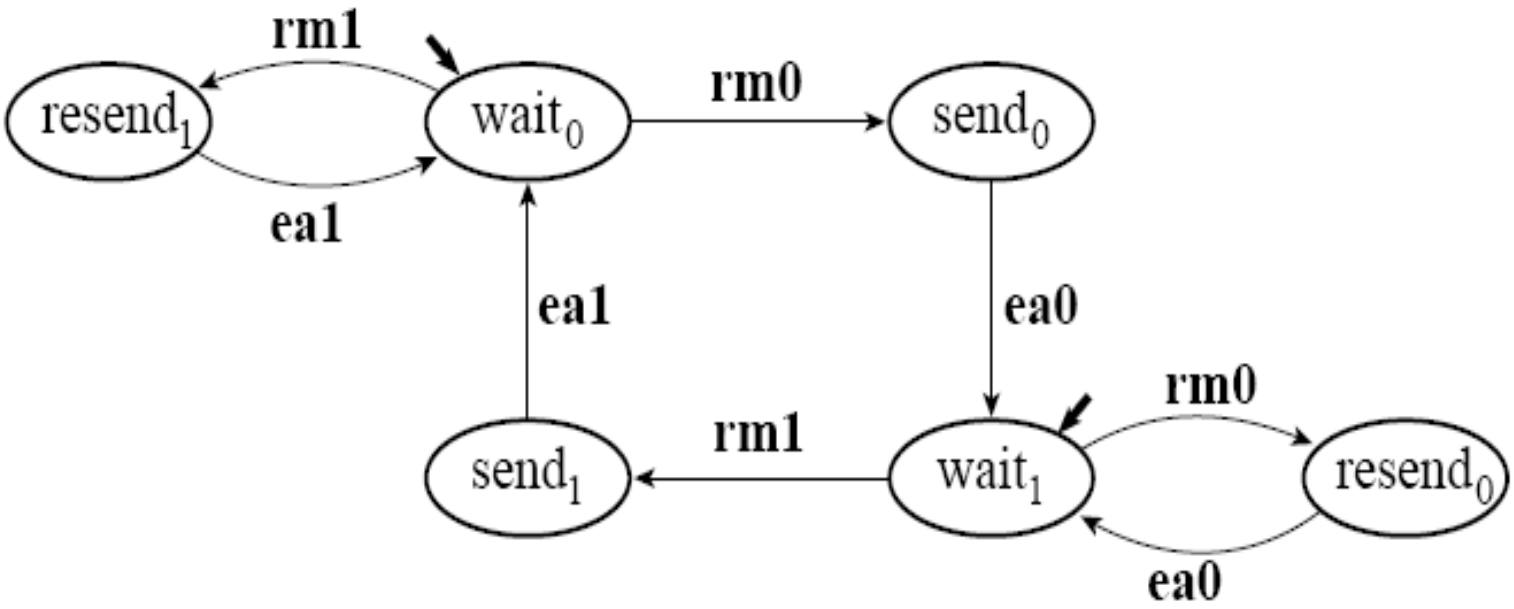
$S_{initial} = \{wait_0, wait_1\}$

$T_{well\_received} = \{t_1', t_3'\}$, az üzenet sikeres fogadása

$T_{ill\_received} = \{t_2', t_4'\}$, hibás üzenettovábbítás



2.13. ábra. Az adó átmeneti rendszer modellje



2.14. ábra. A vevő átmeneti rendszer modellje

Különböző kommunikáció megvalósítása szinkronizációs megszorításokkal:

1. Ha mindkét irányban hibátlan a kommunikáció:

$$I = \{\langle em0, rm0\rangle, \langle em1, rm1\rangle, \langle ra0, ea0\rangle, \langle ra1, ea1\rangle\}$$

2. Ha csak a vevőtől az adóig irányban hibátlan a kommunikáció:

$$I_e = I \cup \{\langle em0, rm1\rangle, \langle em1, rm0\rangle\}$$

3. Ha csak az adótól a vevőig irányban hibátlan a kommunikáció:

$$I_r = I \cup \{\langle ra0, ea1\rangle, \langle ra1, ea0\rangle\}$$

4. Ha mindkét irányban megbízhatatlan a kommunikáció:

$$I_{er} = I_e \cup I_r$$

Jelölések a mindkét irányban megbízhatatlan csatornát feltételező *ABP* protokoll modelljében:

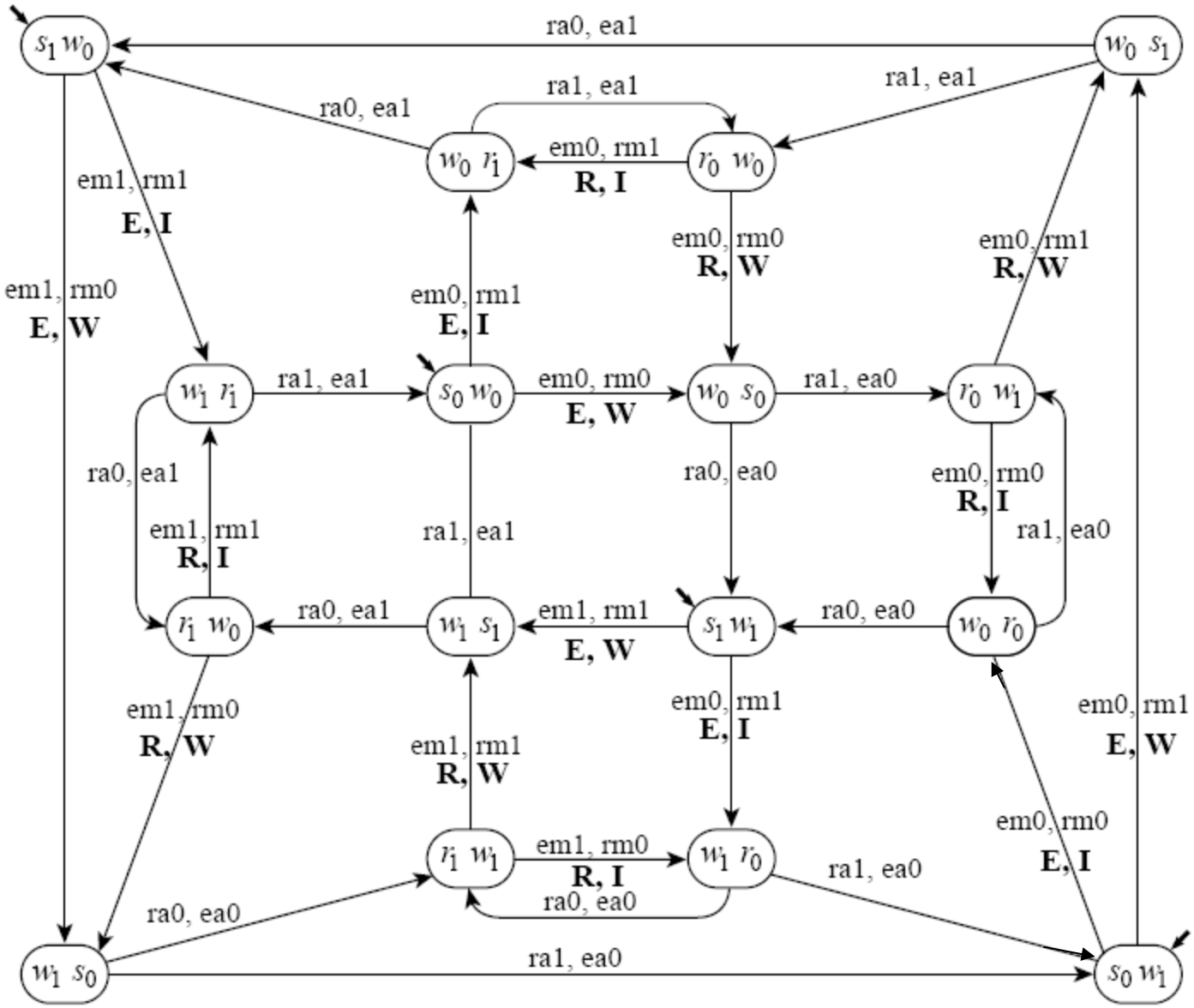
E = (1, *emission*) : az adó komponens az üzenetet először küldi ki;

R = (1, *re-emission*) : az adó komponens az üzenetet ismételten küldi ki;

W = (2, *well_received*): hibamentes üzenetküldések;

I = (2, *ill_received*) : hibás üzenetküldések;

Animáció: az I, I_e, I_r, I_{er} szinkronizációs megszorítások szerinti átmeneti rendszerek szemléltetésére;



2.15. ábra. Az ABP protokoll modellje mindkét irányban megbízhatatlan csatornát feltételezve

Temporális logikák

Átmeneti rendszerek időbeli (események sorrendisége) változásai adhatók meg.

Lineáris temporális logika: az időt, időpillanatok egymást követő sorozataként kezeli.

Elágazó temporális logika: időben egy eseménynek több rákövetkezője lehet.

Nem temporális logika: állapotoknak, átmeneteknek lokális tulajdonsága adható meg.

Kijelentés logika

$\mathcal{A} = (S, T, \alpha, \beta, S_{x_1},\ldots, S_{x_n}, T_{y_1},\ldots, T_{y_m})$ (X, Y)-paraméterezett átmeneti rendszer, $X=\{x_1,\ldots, x_n\}$, $Y=\{y_1,\ldots, y_m\}$.

jelölés:

$AP = \{P_x \mid x \in X\}$, állapot atomi kijelentések halmaza;

$AP_\tau = \{Q_y \mid y \in Y\}$, átmenet atomi kijelentések halmaza;

A logika szimbólumai:

$\underline{0}$, $\underline{1}$ állapot logikai konstansok,

P_x állapot atomi kijelentés, minden $x \in X$-re,

$\wedge$ (logikai és) kétváltozós operátor,

$\neg$ (logikai negáció) egyváltozós operátor;

AP feletti kijelentés logika állapotformulái:

- $\underline{0}$, $\underline{1}$ állapot logikai konstansok,
- a, minden $a \in AP$-re,
- ha f_1 és f_2 állapotformulák, akkor $f_1 \wedge f_2$ állapotformula,
- ha f állapotformula, akkor $\neg f$ állapotformula.

Formulák szemantikája:

legyen f állapotformula, $s \in S$,

$\mathcal{A}, s \models f$ jelentése:

$\mathcal{A}$ átmeneti rendszer s állapotában kielégíti az f állapotformulát;

$\mathcal{A}, s \not\models f$ jelentése:

$\mathcal{A}$ átmeneti rendszer az s állapotában nem elégíti ki az f állapotformulát.

$\mathcal{A}, s \models f$ reláció definiálása az f állapotformula felépítése szerinti indukcióval:

- ha $f = \underline{0}$, akkor $\mathcal{A}, s \not\models f$,
- ha $f = \underline{1}$, akkor $\mathcal{A}, s \models f$,
- ha $f = a$, ahol $a \in AP$ és $a = \mathrm{P}_x$ valamely $x \in X$-re, akkor $\mathcal{A}, s \models f \Leftrightarrow s \in S_x$,
- ha $f = f_1 \wedge f_2$, akkor $\mathcal{A}, s \models f \Leftrightarrow \mathcal{A}, s \models f_1$ és $\mathcal{A}, s \models f_2$,
- ha $f = \neg f'$, akkor $\mathcal{A}, s \models f \Leftrightarrow \mathcal{A}, s \not\models f'$.

A logika állapotformuláival az átmeneti rendszer állapotainak lokális tulajdonsága adható meg.

AP_τ feletti kijelentés logika átmenetformulái:

- $\underline{0}_\tau$, $\underline{1}_\tau$ átmenet logikai konstansok,
- a_τ, minden $a_\tau \in AP_\tau$ -ra,
- ha g_1 és g_2 átmenetformulák, akkor $g_1 \wedge_\tau g_2$ átmenetformula,
- ha g átmenetformula, akkor $\neg_\tau g$ átmenetformula.

Átmenetformulák szemantikája:

legyen g átmenetformula és $t \in T$,

$\mathcal{A}, t \models g$ jelentése:

$\mathcal{A}$ átmeneti rendszer t átmenete kielégíti az g átmenetformulát.

$\mathcal{A}, t \models g$ reláció definiálása az g átmenetformula felépítése szerinti indukcióval:

- ha $g = \underline{0}_\tau$, akkor $\mathcal{A}, t \not\models g$,
- ha $g = \underline{1}_\tau$, akkor $\mathcal{A}, t \models g$,
- ha $g = a_\tau$, ahol $a_\tau \in AP_\tau$ és $a_\tau = Q_y$ valamely $y \in Y$-ra, akkor $\mathcal{A}, t \models g \Leftrightarrow t \in T_y$,
- ha $g = g_1 \wedge_\tau g_2$, akkor $\mathcal{A}, t \models g \Leftrightarrow \mathcal{A}, t \models g_1$ és $\mathcal{A}, t \models g_2$,
- ha $g = \neg_\tau g'$, akkor $\mathcal{A}, t \models g \Leftrightarrow \mathcal{A}, t \not\models g'$.

A logika átmenetformuláival az átmeneti rendszer átmeneteinek lokális tulajdonsága adható meg.

LTL (Linear Temporal Logic)

$\mathcal{A}=(S, T, \alpha, \beta, S_{x_1},\ldots, S_{x_n}, T_{y_1}, \ldots, T_{y_m})$ (X, Y)-paraméteres átmeneti rendszer, $X=\{x_1,\ldots, x_n\}$, $Y=\{y_1,\ldots, y_m\}$, minden állapotnak van rákövetkezője.

LTL szimbólumai:

- az *AP* feletti kijelentés logika szimbólumai,
- **X** (next) egyváltozós temporális operátor,
- **U** (until) kétváltozós temporális operátor.

A logika útformulái:

- $\underline{0}$, $\underline{1}$, a, $\forall a \in AP$-re,
- $f_1 \wedge f_2$, $\neg f$, ha f_1, f_2, f útformulák,
- $\mathbf{X}f$, ha f útformula,
- $f_1 \mathbf{U} f_2$, ha f_1 és f_2 útformulák.

kifejezhető operátorok a logikában:

- $\mathbf{F}f = \underline{1}\mathbf{U}f$, $f_1\mathbf{B}f_2 = \neg((\neg f_1)\mathbf{U}f_2)$, $\mathbf{G}f = \neg(\mathbf{F}(\neg f))$,
- $\mathbf{G}^{\infty}f = \mathbf{FG}f$, $\mathbf{F}^{\infty}f = \mathbf{GF}f$
- $f_1\mathbf{W}f_2 = (f_1\mathbf{U}f_2)\vee\mathbf{G}f_1$ (gyenge until).

F helyett a ◊ és **G** helyett a [] vagy a □ jelölés is gyakori.

jelölés:

Legyen $c = s_0, s_1,\ldots$ végtelen út az $\mathscr{A}$-ban, ahol $\forall i \geq 0$-ra $s_i \in S$. $\forall\ i \geq 0$-ra $c^i = s_i , s_{i+1},\ldots$.

Formulák szemantikája:

$\mathscr{A}, c \models f$ reláció definiálása f útformula felépítése szerint:

- ha $f = \underline{0}$, akkor $\mathscr{A}, c \not\models f$,
- ha $f = \underline{1}$, akkor $\mathscr{A}, c \models f$,

- $f = a$, ahol $a = P_x \in AP$ valamely $x \in X$-re:

  $\mathcal{A}, c \models f \Leftrightarrow s_0 \in S_x$,

- $f = f_1 \wedge f_2$: $\mathcal{A}, c \models f \Leftrightarrow \mathcal{A}, c \models f_1$ és $\mathcal{A}, c \models f_2$,

- $f = \neg f'$: $\mathcal{A}, c \models f \Leftrightarrow \mathcal{A}, c \not\models f'$,

- $f = \mathbf{X} f'$: $\mathcal{A}, c \models f \Leftrightarrow \mathcal{A}, c^1 \models f'$,

- $f = f_1 \mathbf{U} f_2$: $\mathcal{A}, c \models f \Leftrightarrow \exists\, j \geq 0$, hogy $\mathcal{A}, c^j \models f_2$, $\forall 0 \leq i < j$-re

  $\mathcal{A}, c^i \models f_1$.

kifejezhető operátorok szemantikája:

$\mathcal{A}, c \models \mathbf{F} f \Leftrightarrow \exists\, i \geq 0$, hogy $\mathcal{A}, c^i \models f$

$\mathcal{A}, c \models \mathbf{G} f \Leftrightarrow \forall\, i \geq 0$, hogy $\mathcal{A}, c^i \models f$

$\mathcal{A}, c \models \mathbf{F}^{\infty} f \Leftrightarrow \forall$ i ≥ 0-ra $\exists$ j $\geq$ i, hogy $\mathcal{A}, c^j \models f$

$\mathcal{A}, c \models \mathbf{G}^{\infty} f \Leftrightarrow \exists$ i ≥ 0, hogy $\forall$ j $\geq$ i-re $\mathcal{A}, c^j \models f$

$\mathcal{A}, c \models f_1 \mathbf{B} f_2 \Leftrightarrow \forall$ j ≥ 0-ra, ha $\mathcal{A}, c^j \models f_2$, akkor $\exists$ i<j, hogy $\mathcal{A}, c^i \models f_1$.

Legyen $s \in S$, $S_0 \subseteq S$ a kezdőállapotok halmaza, f *AP* feletti LTL formula.

jelölések:

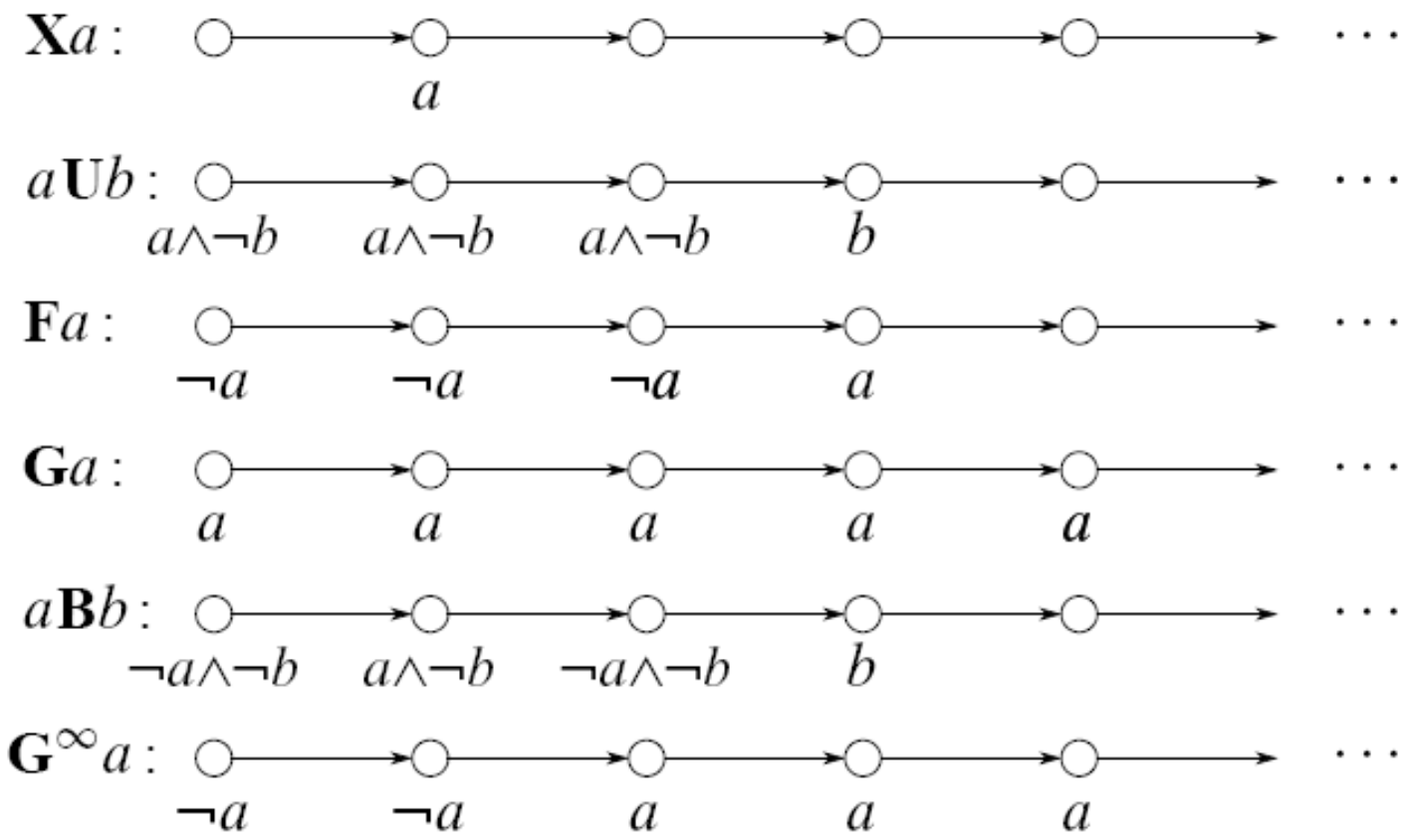
$\mathcal{A}$ kielégíti az s állapotban az f LTL formulát: $\mathcal{A}, s \models f$;

$\mathcal{A}$ kielégíti az f LTL formulát: $\mathcal{A} \models f$;

$\mathcal{A}, s \models f$ *és* $\mathcal{A} \models f$ relációk definiálása:

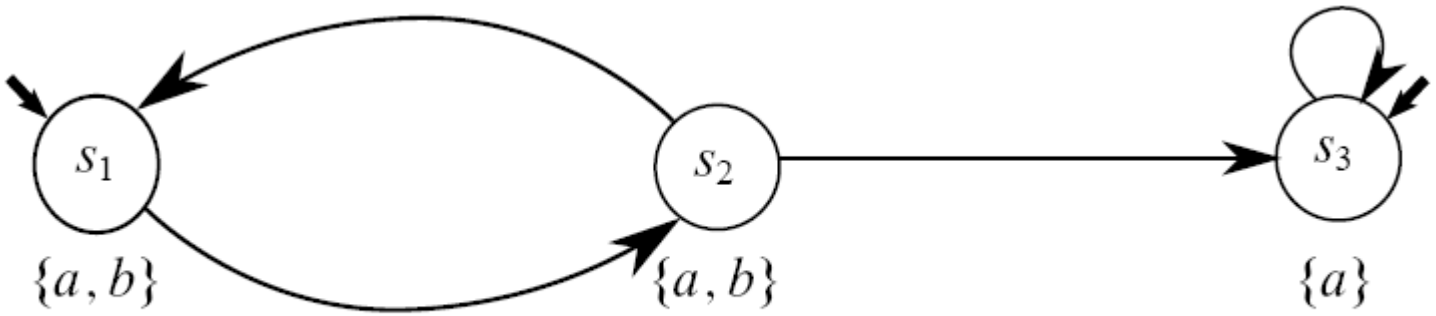
$\mathcal{A}, s \models f \Leftrightarrow \forall c$ végtelen s-ből induló útra $\mathcal{A}, c \models f$;

$\mathcal{A} \models f \Leftrightarrow \forall s \in S_0$-ra $\mathcal{A}, s \models f$;



3.1. ábra. Példák LTL formulákat kielégítő egy-egy útra

példa:



3.2. ábra. $\{a, b\}$ állapot-paraméteres $\mathcal{A}$ átmeneti rendszer

$\mathcal{A}, s_i \models \mathbf{G}(a)$ minden $i = 1,2,3$-ra, $\quad \mathcal{A} \not\models \mathbf{X}(a \wedge b)$,

$\mathcal{A}, s_1 \models \mathbf{X}(a \wedge b)$, $\quad \mathcal{A} \models \mathbf{G}(\neg b \rightarrow \mathbf{G}(a \wedge \neg b))$,

$\mathcal{A}, s_2 \not\models \mathbf{X}(a \wedge b)$, $\quad \mathcal{A} \not\models b\mathbf{U}(a \wedge \neg b)$.

$\mathcal{A}, s_3 \not\models \mathbf{X}(a \wedge b)$,

Átmenet-paraméteres rendszerekre az LTL definiálása

Legyen $c = t_0, t_1, \ldots$ végtelen út az $\mathscr{A}$-ban, ahol $\forall i \geq 0$-ra $t_i \in T$, g LTL útformula az AP_τ átmenet atomi kijelentések felett.

$c^i = t_i, t_{i+1}, \ldots$ végtelen út.

$\mathscr{A}, c \models g$ definiálása:

- ha $g = \underline{0}_\tau$, akkor $\mathscr{A}, c \not\models g$,
- ha $g = \underline{1}_\tau$, akkor $\mathscr{A}, c \models g$,
- ha $g = a_\tau$, ahol $a_\tau = Q_y \in AP_\tau$ valamely $y \in Y$-ra, akkor $\mathscr{A}, c \models g \Leftrightarrow t_0 \in T_y$,
- ha $g = g_1 \wedge_\tau g_2$, akkor $\mathscr{A}, c \models g \Leftrightarrow \mathscr{A}, c \models g_1$ és $\mathscr{A}, c \models g_2$,
- ha $g = \neg_\tau g'$, akkor $\mathscr{A}, c \models g \Leftrightarrow \mathscr{A}, c \not\models g'$,

- ha $g = \mathbf{X}_\tau g'$, akkor $\mathscr{A}, c \models g \Leftrightarrow \mathscr{A}, c^1 \models g'$,
- ha $g = g_1 \mathbf{U}_\tau g_2$, akkor $\mathscr{A}, c \models g \Leftrightarrow \exists\, j \geq 0$, hogy $\mathscr{A}, c^j \models g_2$, $\forall 0 \leq i < j$-re $\mathscr{A}, c^i \models g_1$.

HML (**H**ennessy-**M**ilner **L**ogic)

A véges címkehalmaz feletti HML szimbólumai:

- $\underline{0}$, $\underline{1}$ logikai konstansok,
- $\wedge$ (logikai és) kétváltozós operátor,
- $\neg$ (logikai negáció) egyváltozós operátor,
- $\langle \boldsymbol{a} \rangle$ egyváltozós operátorok $\forall a \in A$-ra.

HML állapotformulái:

- $\underline{0}$, $\underline{1}$ logikai konstansok,
- $f_1 \wedge f_2$, ha f_1 és f_2 állapotformulák,
- $\neg f$, ha f állapotformula,
- $\langle \boldsymbol{a} \rangle f$, ha f állapotformula, $a \in A$.

Kifejezhető operátor:

$$[\boldsymbol{a}]f = \neg \langle \boldsymbol{a} \rangle \neg f$$

Legyen $\mathscr{A} = (S, T, \alpha, \beta, \lambda)$ egy A feletti címkézett átmeneti rendszer, $s \in S$, $a \in A$, f HML formula, S_0 az $\mathscr{A}$ kezdőállapotainak halmaza.

$\mathscr{A}, s \models f$ definiálása:

- ha $f = \underline{0}$, akkor $\mathscr{A}, s \not\models f$,
- ha $f = \underline{1}$, akkor $\mathscr{A}, s \models f$,
- ha $f = f_1 \wedge f_2$, akkor $\mathscr{A}, s \models f \Leftrightarrow \mathscr{A}, s \models f_1$ és $\mathscr{A}, s \models f_2$,
- ha $f = \neg f'$, akkor $\mathscr{A}, s \models f \Leftrightarrow \mathscr{A}, s \not\models f'$
- ha $f = \langle \boldsymbol{a} \rangle f'$, akkor $\mathscr{A}, s \models f \Leftrightarrow \exists t \in T$, $\alpha(t)=s$, $\lambda(t)=a$, $\mathscr{A}, \beta(t) \models f'$

$[\boldsymbol{a}]f$ szemantikája:

$\mathscr{A}, s \models [\boldsymbol{a}]f \Leftrightarrow \forall t \in T$, $\alpha(t)=s$, $\lambda(t)=a$, akkor $\mathscr{A}, \beta(t) \models f$;

$\mathscr{A} \models f \Leftrightarrow \forall s \in S_0$-ra $\mathscr{A}, s \models f$;

HML kiterjesztései:

- $\mathscr{A}$ címkézett és $(X,\varnothing)$-paraméteres átmeneti rendszer:
  AP elemei állapot atomi kijelentések a logikában.
- $\mathscr{A}$ $(\varnothing,Y)$-paraméteres átmeneti rendszer:

HML állapotformulái:

- $\underline{0}$, $\underline{1}$ logikai konstansok,
- $f_1 \wedge f_2$, $\neg f$,ha f_1 , f_2 és f állapotformulák,
- $\langle \boldsymbol{g} \rangle f$, ha f állapotformula, g átmenetformula.

$\langle g \rangle f$ szemantikája:

$\mathscr{A}, s \models \langle \boldsymbol{g} \rangle f \Leftrightarrow \exists t \in T, \alpha(t)=s, \mathscr{A}, t \models g, \mathscr{A}, \beta(t) \models f;$

$[\boldsymbol{g}] f$ szemantikája:

$\mathscr{A}, s \models [\boldsymbol{g}] f \Leftrightarrow \forall t \in T, \alpha(t)=s, \mathscr{A}, t \models g,$ akkor $\mathscr{A}, \beta(t) \models f;$

Dicky logika

(X, Y)-paraméteres átmeneti rendszerek tulajdonságai adhatók meg vele.

A logika szimbólumai:

- $\underline{0}_\sigma$, $\underline{1}_\sigma$, $\underline{0}_\tau$, $\underline{1}_\tau$ állapot, illetve átmenet logikai konstansok,
- P_x állapot atomi kijelentés, minden $x \in X$-re,
- Q_y átmenet atomi kijelentés, minden $y \in Y$-ra,
- $\wedge_\sigma$, $\vee_\sigma$, $-_\sigma$, $\wedge_\tau$, $\vee_\tau$, $-_\tau$ kétváltozós állapot, illetve átmenet operátorok,
- ***src***, ***tgt***, ***in***, ***out*** egyváltozós állapot, illetve átmenet operátorok;

állapotformulák:

$\underline{0}_\sigma$, $\underline{1}_\sigma$, P_x $\forall x \in X$-re,

$f_1 \wedge_\sigma f_2$, $f_1 \vee_\sigma f_2$, $f_1 -_\sigma f_2$, ha f_1 ,f_2 állapotformulák,

src(g), ***tgt***(g), ha g átmenetformula.

átmenetformulák:

$\underline{0}_\tau$, $\underline{1}_\tau$, Q_y $\forall y \in Y$-ra,

$g_1 \wedge_\tau g_2$, $g_1 \vee_\tau g_2$, $g_1 -_\tau g_2$, ha g_1, g_2 átmenetformulák,

in(f), ***out***(f), ha f állapotformula.

Legyen $\mathcal{A}=(S, T, \alpha, \beta, S_{x_1},\ldots, S_{x_n}, T_{y_1}, \ldots, T_{y_m})$ (X,Y)-paraméteres átmeneti rendszer, $X=\{x_1,\ldots,x_n\}$, $Y=\{y_1,\ldots, y_m\}$, $s \in S$, $t \in T$, f, f_1, f_2, állapotformulák, g, g_1, g_2 átmenetformulák.

formulák szemantikája*:*

$\underline{0}_\sigma$, $\underline{1}_\sigma$, $\underline{0}_\tau$, $\underline{1}_\tau$, P_x $\forall x \in X$-re, Q_y $\forall y \in Y$-ra, vagyis a logika atomi formuláinak és a $\wedge_\sigma$, $\vee_\sigma$, $\wedge_\tau$, $\vee_\tau$ operátorok szemantikája, mint a kijelentés logikában.

$\mathcal{A}, s \models f_1 -_\sigma f_2 \Leftrightarrow \mathcal{A}, s \models f_1$ és $\mathcal{A}, s \not\models f_2$,

$\mathcal{A}, s \models \mathbf{src}(g) \Leftrightarrow \exists t \in T, \alpha(t)=s, \mathcal{A}, t \models g$,

$\mathcal{A}, s \models \mathbf{tgt}(g) \Leftrightarrow \exists t \in T, \beta(t) =s, \mathcal{A}, t \models g$,

$\mathcal{A}, t \models g_1 -_\tau g_2 \Leftrightarrow \mathcal{A}, t \models g_1$ és $\mathcal{A}, t \not\models g_2$,

$\mathcal{A}, t \models \mathbf{in}(f) \Leftrightarrow \mathcal{A}, \beta(t) \models f$,

$\mathcal{A}, t \models \mathbf{out}(f) \Leftrightarrow \mathcal{A}, \alpha(t) \models f$.

negáció ($\neg$) kifejezése:

$\mathscr{A}, s \models \neg f \Leftrightarrow \mathscr{A}, s \models \underline{1}_\sigma -_\sigma f,$

$\mathscr{A}, t \models \neg g \Leftrightarrow \mathscr{A}, t \models \underline{1}_\tau -_\tau g.$

HML $\langle a \rangle f$ kifejezése:

$\mathscr{A}, s \models \langle \boldsymbol{a} \rangle f \Leftrightarrow \mathscr{A}, s \models \boldsymbol{src}(\mathrm{Q}_a \wedge_\tau \boldsymbol{in}(f)).$

példák:

$\mathscr{A}, s \models \boldsymbol{src}(\underline{1}_\tau) \Leftrightarrow$ s állapotból indul ki átmenet

$\mathscr{A}, s \models \underline{1}_\sigma -_\sigma \boldsymbol{tgt}(\underline{1}_\tau) \Leftrightarrow$ s állapotba nem vezet átmenet

$\mathscr{A}, s \models \mathrm{P}_{x_1} \wedge_\sigma \boldsymbol{src}(\underline{1}_\tau \wedge_\tau \boldsymbol{in}(\mathrm{P}_{x_2})) \Leftrightarrow \; s \in S_{x_1}, \exists t \in T, \beta(t) \in S_{x_2}.$

$\mathscr{A}, t \models \boldsymbol{out}(\mathrm{P}_x) \Leftrightarrow \alpha(t) \in S_x$

$\mathscr{A}, t \models \boldsymbol{in}(\mathrm{P}_x \wedge_\sigma \boldsymbol{src}(\mathrm{Q}_y)) \Leftrightarrow \beta(t) \in S_x, \exists t' \in T, \beta(t) = \alpha(t'), t' \in T_y$

CTL (Computational Tree Logic)

Legyen $\mathscr{A} = (S, T, \alpha, \beta, S_{x_1},\ldots, S_{x_n})$ $(X, \varnothing)$-*p*araméterezett átmeneti rendszer, $X=\{x_1,\ldots, x_n\}$, $S_0 \subseteq S$ a kezdőállapotok halmaza, $s_0 \in S$, minden állapotnak van rákövetkezője.

s_0 gyökerű számítási fa olyan irányított, címkézett végtelen fa, melynek, ha valamely $s \in S$ csúcsa és s-ből T-beli átmenetekkel $s_1,\ldots, s_k$ állapotok elérhetők $\mathscr{A}$-ban, akkor az s címkéjű csúcsnak k darab leszármazottja van, ezek címkéi $s_1,\ldots, s_k$.

AP feletti CTL szimbólumai:

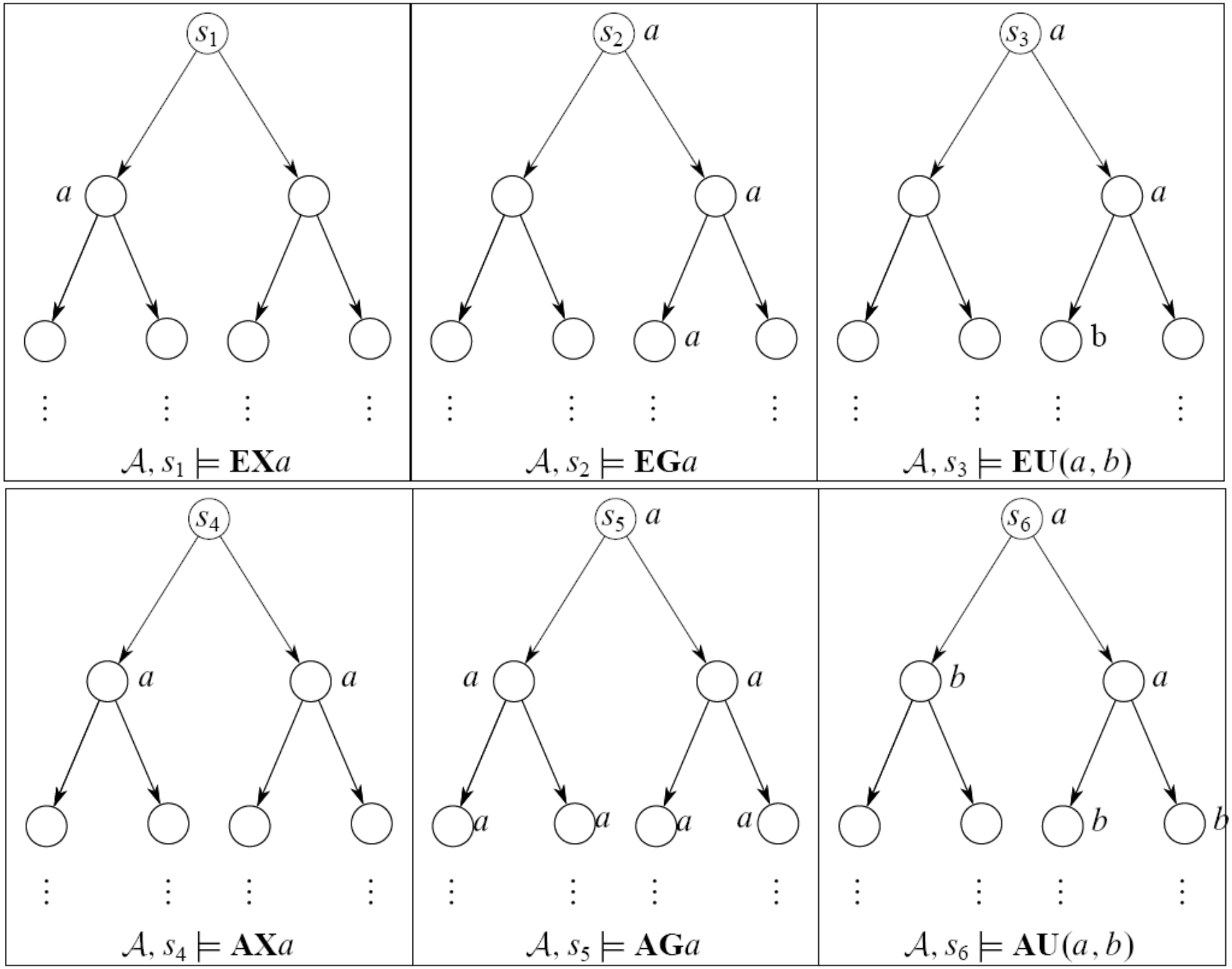
- *AP* feletti kijelentés logika szimbólumai,
- **EX**, **AX** egyváltozós operátorok,
- **EU**, **AU** kétváltozós operátorok.

állapotformulák:

- $\underline{0}$, $\underline{1}$, a $\forall a \in AP$-re,
- $f_1 \wedge f_2$, $\neg f$, ahol f_1, f_2, f állapotformulák,
- **EX**f, **AX**f, ahol f állapotformula,
- **EU**(f_1, f_2), **AU**(f_1, f_2), ahol f_1, f_2 állapotformulák.

Az állapotformulák szemantikája:

- $\underline{0}$, $\underline{1}$, a $\forall a \in AP$-re, $f_1 \wedge f_2$, $\neg f$ esetén, mint a kijelentés logikában,
- $\mathcal{A}, s \models \mathbf{EX}f \Leftrightarrow \exists t \in T, \alpha(t)=s, \mathcal{A}, \beta(t) \models f,$
- $\mathcal{A}, s \models \mathbf{AX}f \Leftrightarrow \forall\ t \in T, \alpha(t)=s, \mathcal{A}, \beta(t) \models f,$
- $\mathcal{A}, s \models \mathbf{EU}(f_1, f_2) \Leftrightarrow \exists\ c=s_0, s_1, \ldots$ végtelen út az $\mathcal{A}$-ban, $\alpha(c)=s$, $\exists\ j \geq 0$, hogy $\mathcal{A}, s_j \models f_2$, $\forall 0 \leq i < j$-re $\mathcal{A}, s_i \models f_1$,
- $\mathcal{A}, s \models \mathbf{AU}(f_1, f_2) \Leftrightarrow \forall\ c=s_0, s_1, \ldots$ végtelen út az $\mathcal{A}$-ban, $\alpha(c)=s$, $\exists\ j \geq 0$, hogy $\mathcal{A}, s_j \models f_2$, $\forall 0 \leq i < j$-re $\mathcal{A}, s_i \models f_1$.


s_1
a
$\mathcal{A}, s_1 \models \mathbf{EX}a$
s_2 a
a
a
$\mathcal{A}, s_2 \models \mathbf{EG}a$
s_3 a
a
b
$\mathcal{A}, s_3 \models \mathbf{EU}(a, b)$
s_4
a
a
$\mathcal{A}, s_4 \models \mathbf{AX}a$
s_5 a
a
a
a
a
a
$\mathcal{A}, s_5 \models \mathbf{AG}a$
s_6 a
b
a
b
b
$\mathcal{A}, s_6 \models \mathbf{AU}(a, b)$

egyéb kifejezhető operátorok:

$\mathbf{EF}f = \mathbf{EU}(\underline{1}, f)$, $\mathbf{AF}f = \mathbf{AU}(\underline{1}, f)$,

$\mathbf{EG}f = \neg(\mathbf{AF}\ (\neg f))$, $\mathbf{AG}f = \neg(\mathbf{EF}\ (\neg f))$.

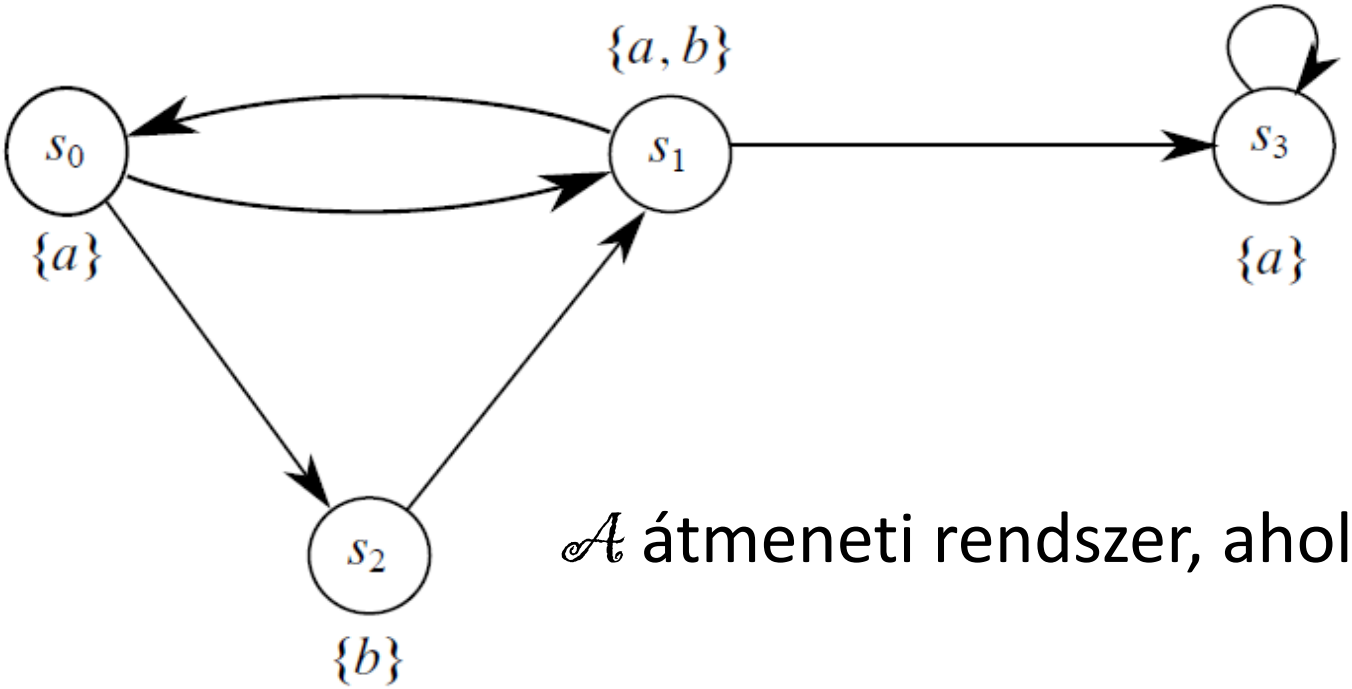
jelölés:

Legyen f CTL formula, $S_f = \{s \in S \mid \mathcal{A}, s \models f\}$.

Animáció:

$\mathcal{A}$ átmeneti rendszer kielégíti az f formulát reláció definiálása:

$$\mathcal{A} \models f \Leftrightarrow S_0 \subseteq S_f.$$



$\mathcal{A}$ átmeneti rendszer, ahol $AP = \{a, b\}$

$f = \mathbf{EX}a$: $S_f = \{s_0, s_1, s_2, s_3\}$, $f = \mathbf{AX}a$: $S_f = \{s_1, s_2, s_3\}$,

$f = \mathbf{EG}a$: $S_f = \{s_0, s_1, s_3\}$, $f = \mathbf{AG}a$: $S_f = \{s_3\}$,

$f = \mathbf{EF}(\mathbf{EG}a)$: $S_f = \{s_0, s_1, s_2, s_3\}$, $f = \mathbf{AU}(a, b)$: $S_f = \{s_0, s_1, s_2\}$,

$f = \mathbf{AU}(\neg a, b)$: $S_f = \{s_1, s_2\}$, $f = \neg a \wedge \mathbf{AU}(\neg a, b)$: $S_f = \{s_2\}$,

$f = \mathbf{EU}(a, (\neg a \wedge \mathbf{AU}(\neg a, b)))$: $S_f = \{s_0, s_1, s_2\}$.

CTL*

Legyen $\mathcal{A} = (S, T, \alpha, \beta, S_{x_1}, \ldots, S_{x_n})$ $(X, \varnothing)$-paraméteres átmeneti rendszer, $X=\{x_1, \ldots, x_n\}$, $S_0 \subseteq S$ a kezdőállapotok halmaza, $s \in S$, minden állapotnak van rákövetkezője.

A logika szimbólumai:

- AP feletti kijelentés logika szimbólumai,
- **E** (egzisztenciális) útkvantor,
- **X**, **U** lineáris temporális operátorok.

A logika állapotformulái:

- $\underline{0}$, $\underline{1}$, a $\forall a \in AP$-re,
- $f_1 \wedge f_2$, $\neg f$, ahol f_1, f_2, f állapotformulák,
- **E**g, ahol g útformula.

A logika útformulái:

- minden állapotformula,
- $g_1 \wedge g_2$, $\neg g$, ahol g_1, g_2, g útformulák,
- $\mathbf{X}g$, ahol g útformula,
- $\mathbf{U}(g_1, g_2)$, ahol g_1 , g_2 útformulák.

Az állapotformulák szemantikája:

- $\underline{0}$, $\underline{1}$, a $\forall a \in AP$-re, $f_1 \wedge f_2$, $\neg f$ állapotformuláknál, mint a kijelentés logikában,
- $\mathscr{A}, s \models \mathbf{E}g \Leftrightarrow \exists\, c = s_0, s_1, \ldots$ végtelen út, $s = s_0$, $\mathscr{A}, c \models g$.

Az útformulák szemantikája:

$c = s_0, s_1, \ldots$ végtelen út az $\mathcal{A}$ átmeneti rendszerben.

- ha $g=f$, ahol f állapotformula, akkor $\mathcal{A}, c \models g \Leftrightarrow \mathcal{A}, s_0 \models f$,
- $g_1 \wedge g_2$, $\neg g$, $\mathbf{X}g$, $\mathbf{U}(g_1, g_2)$, ahol g_1, g_2, g útformulák szemantikája, mint az LTL-ben.

jelölés:

f állapotformula CTL*-ban, ekkor $S_f = \{s \in S \mid \mathcal{A}, s \models f\}$.

$\mathcal{A}$ kielégíti az f állapotformulát reláció definiálása:

$$\mathcal{A} \models f \Leftrightarrow S_0 \subseteq S_f.$$

Univerzális útkvantor:

$$\mathbf{A}g = \neg(\mathbf{E}(\neg g))$$

LTL, CTL, CTL* logikák kifejezőerejének összehasonlítása

Logika kifejezőereje:

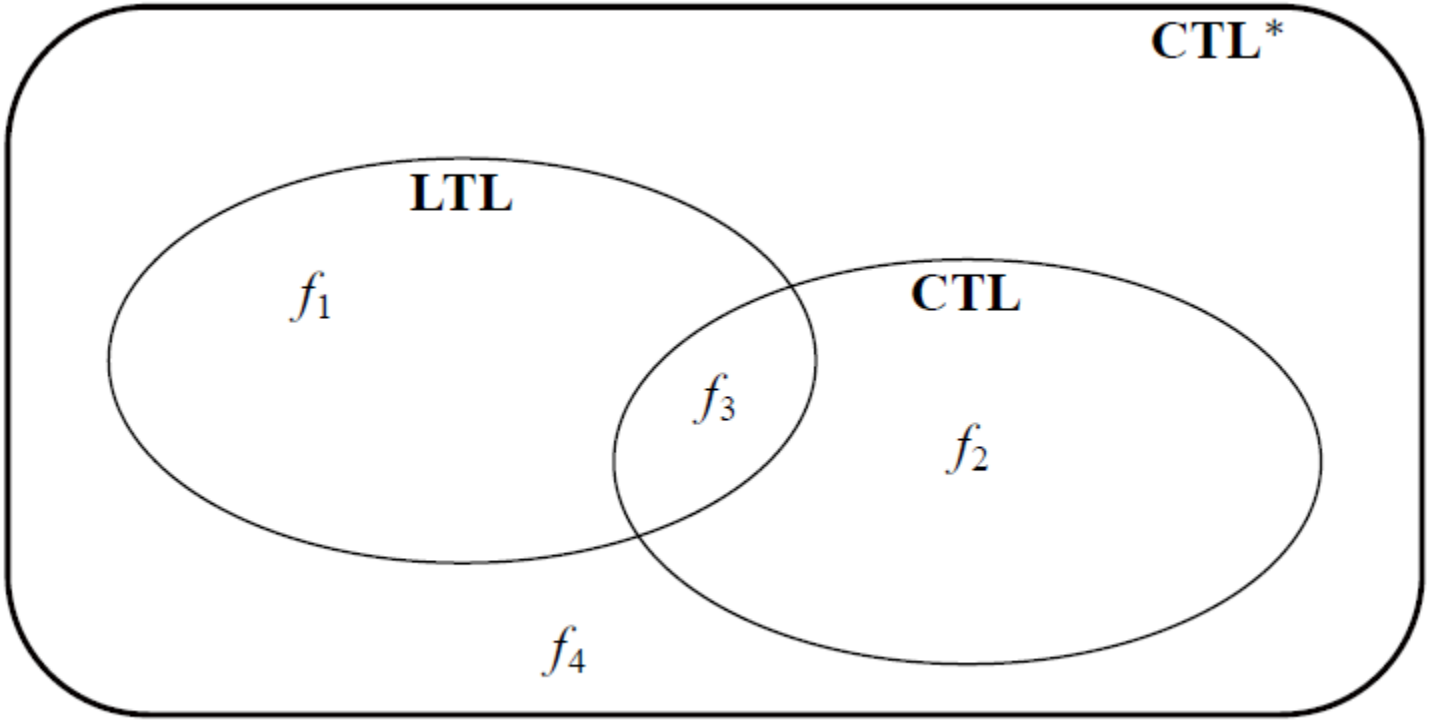
a logikában kifejezhető tulajdonságok összessége.

Legyen f, g AP atomi kijelentések feletti LTL, CTL,CTL* állapotformulák.

f, g állapotformulák ekvivalensek ($f \equiv g$), ha bármely AP feletti $\mathcal{A}$ átmeneti rendszerre és $\forall\, s \in S$-re teljesül

$$\mathcal{A}, s \models f \Leftrightarrow \mathcal{A}, s \models g.$$

- ha f LTL formula, akkor $\mathbf{A}f$ CTL* és $f \equiv \mathbf{A}f$;
- minden f CTL formula CTL* formula;



3.5. ábra. Az LTL, CTL és CTL logikák közötti kapcsolat*

f_1=**FG***a*, f_2=**AG**(**EF***a*), f_3=**GF***a*≡ **AGAF***a*, f_4=**AFG***a*∨**AG**(**EF***a*)

- LTL és CTL kifejezőereje nem összehasonlítható.
- CTL* kifejezőereje nagyobb az LTL és a CTL kifejezőerejénél.

Időzített átmeneti rendszerek

Legyen A egy ábécé, $A' = A \cup \{\varepsilon(d) \mid d \in \mathbb{R}_{\geq 0}\}$.
A' feletti (valós idejű) időzített átmeneti rendszer olyan $\mathcal{A} = (S, T, \alpha, \beta, \lambda)$ címkézett átmeneti rendszert ($\lambda: T \to A'$), melyre teljesülnek az alábbiak:

(1) $\forall s \in S : s \xrightarrow{\varepsilon(0)} s;$

(2) $\forall s \in S, \forall d, d' \in \mathbb{R}_{\geq 0}:$
ha $s \xrightarrow{\varepsilon(d)} s'$ és $0 < d' < d$, akkor $\exists s'' \in S : s \xrightarrow{\varepsilon(d')} s'' \xrightarrow{\varepsilon(d-d')} s';$

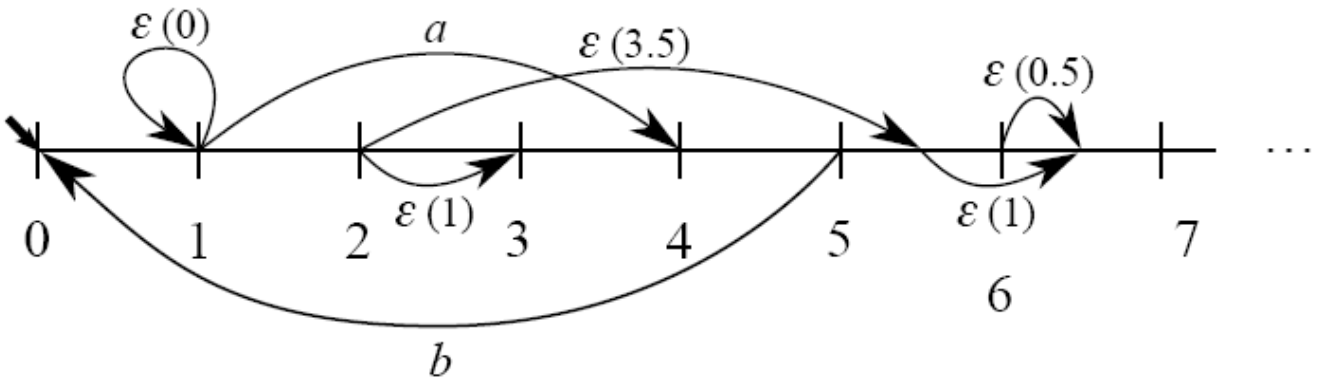
(3) $\forall s, s', s'' \in S, \forall d \in \mathbb{R}_{\geq 0}:$ ha $s \xrightarrow{\varepsilon(d)} s'$ és $s \xrightarrow{\varepsilon(d)} s''$, akkor $s' = s''$.

példa:

$A = \{a, b\}$ címkehalmaz, $S = \mathbb{R}_{\geq 0}$,

$T = \{ 1 \xrightarrow{a} 4 \ , \ \ 5 \xrightarrow{b} 0 \ \} \cup \{ r \xrightarrow{\varepsilon(d)} r+d \ \ | \forall\ r, d \in \mathbb{R}_{\geq 0} \}$

Kezdőállapot: 0



2.16. ábra. A példában definiált időzített átmeneti rendszer néhány átmenete

Időzített automaták

Legyen $C = \{x, y, \ldots\}$ egy halmaz, az *órák halmaza*.

C feletti órafeltételek halmaza az a legszűkebb $\mathcal{B}(C)$ halmaz, melyre $\forall x \in C$, $\lozenge \in \{<, \leq, =, \geq, >\}$, $n \in \mathbb{N}$ esetén $x \lozenge n \in \mathcal{B}(C)$,

$\forall\ g_1, g_2 \in \mathcal{B}(C)$ esetén $g_1 \wedge g_2 \in \mathcal{B}(C)$.

Óraértékelés v: $C \to \mathbb{R}_{\geq 0}$

Órák értékét megváltoztató műveletek:

szünet (delay), *újraindítás* (reset);

jelölések*:*

Legyen *C* egy órahalmaz, *v* egy óraértékelés, $R \subseteq C$.

$v+d\ (x) = v(x) + d$, $\forall\ x \in C$, $d \in \mathbb{R}_{\geq 0}$;

$$v[R \mapsto 0](x) = \begin{cases} 0, \text{ ha } x \in R & \forall\ x \in C; \\ v(x), \text{ különben} \end{cases}$$

megjegyzés:

- ha $R = \{x\}$, akkor R helyett x-t írunk ($v[x \mapsto 0]$);
- $\mathcal{B}(C)$ elemei hely-invariánsként, illetve őrfeltételként jelenhetnek meg az időzített automatában;

Óraértékelések és órafeltételek közötti kielégítési relációt (jelölése $v \models g$, ahol v óraértékelés, g órafeltétel) az órafeltételek felépítése szerinti indukcióval definiáljuk:

$v \models x \lozenge n \Leftrightarrow v(x) \lozenge n$, $\forall x \in C$, $n \in \mathbb{N}$ esetén,

$v \models g_1 \wedge g_2 \Leftrightarrow v \models g_1$ és $v \models g_2$, $\forall g_1, g_2 \in \mathcal{B}(C)$ esetén.

$v \not\models g$ jelöli, ha $v \models g$ reláció nem teljesül;

Órafeltételek ekvivalenciája

$\forall g_1, g_2$ órafeltételre g_1 ekvivalens g_2-vel $\Leftrightarrow$

ha minden v óraértékelésre $v \models g_1 \Leftrightarrow v \models g_2$;

példa:

$C = \{x, y\}$, $v = [x = 1.2, y = 3.1]$

$$v \models x > 1 \wedge x \leq 2,$$

$$v \models x > 0 \wedge y \geq 3,$$

$$v \not\models y \geq 3 \wedge x \leq 1.$$

Legyen C órahalmaz, A az akciók (véges vagy végtelen) halmaza.

A és C feletti időzített automata (L, ℓ_0, E, I), ahol

L a *helyek* véges halmaza,

$\ell_0 \in L$ a *kezdőhely*,

$E \subseteq L \times \mathcal{B}(C) \times A \times \mathcal{P}(C) \times L$ az *átmenetek* (vagy élek) véges halmaza,

$I : L \to \mathcal{B}(C)$ a helyekhez *hely-invariánsok*at rendelő függvény.

jelölés:

$$\ell \xrightarrow{g,a,R} \ell'$$

(ℓ, g, a, R, ℓ') átmenet, ahol ℓ *forráshely*, ℓ' *célhely*, a *akció*, g *őrfeltétel*, R *újraindítandó órák* halmaza.

Időzített automaták grafikus reprezentálása:
a helyeket körök reprezentálják, a helyinvariánsok a hely körben elhelyezve;
az átmeneteket nyilak reprezentálják, nyíl kezdetén az őrfeltétel, közepén az akció, végén az újraindítandó órák (*x*:=0 típusú jelöléssel).

megjegyzés*:*
Az órafeltételekben egész számok szerepelhetnek, de ez nem jelent megszorítást. Ha a valóságban racionális számok szerepelnek az automatában, ezeket a legkisebb közös többszörösükkel végig szorozva, az eredetivel ekvivalens modell kapható (véges sok átmenet lehet az automatákban).

példa:

$C = \{x\}$, $L = \{\text{Off}, \text{Light}, \text{Bright}\}$, $\ell_0 = \text{Off}$,

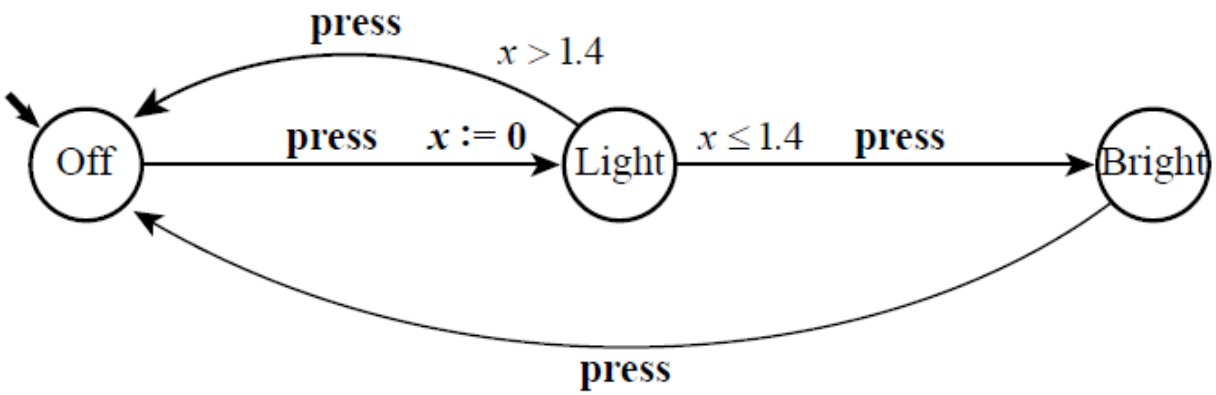
$I(\text{Off}) = I(\text{Light}) = I(\text{Bright}) = \uparrow$, ahol $\uparrow$ az azonosan igaz,

$$E = \{\text{Off} \xrightarrow{\uparrow,\text{press},\{x\}} \text{Light},$$

$$\text{Light} \xrightarrow{x>14,\text{press},\emptyset} \text{Off},$$

$$\text{Light} \xrightarrow{x\leq 14,\text{press},\emptyset} \text{Bright},$$

$$\text{Bright} \xrightarrow{\uparrow,\text{press},\emptyset} \text{Off}\}$$



2.17. ábra. Az érintőkapcsolóval vezérelt lámpa időzített automata modellje

Szemantika (az automata működése)

$\mathcal{A} = (L, \ell_0, E, I)$ időzített automata akciók A és órák C halmaza felett.

$\mathcal{A}$ automatához rendelt időzített átmeneti rendszer

$A' = A \cup \{\varepsilon(d) \mid d \in \mathbb{R}_{\geq 0}\}$ feletti $T(\mathcal{A})=(S, T, \alpha, \beta, \lambda)$, ahol

$S = \{(\ell, v) \mid \ell \in L, v : C \to \mathbb{R}_{\geq 0}, v \models I(\ell)\}$,

T elemei $\forall (\ell, v), (\ell', v') \in S, d \in \mathbb{R}_{\geq 0}, a \in A$ esetén

$(\ell, v) \xrightarrow{a} (\ell', v') \in T \Leftrightarrow \exists \ell \xrightarrow{g,a,R} \ell' \in E, v \models g, v'=v[R \to 0], v' \models I(\ell')$,

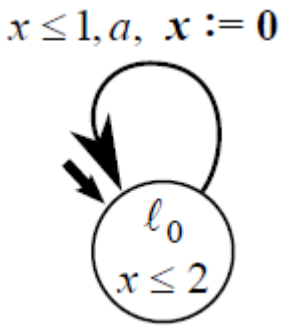
$(\ell, v) \xrightarrow{\varepsilon(d)} (\ell, v+d) \in T \Leftrightarrow v \models I(\ell), v+d \models I(\ell)$.

Van egy *kezdő* nevű állapotparaméter,

$S_{kezdő} = \{(\ell_0, v_0)\}$, ahol $v_0(x) = 0$, $\forall x \in C$-re.

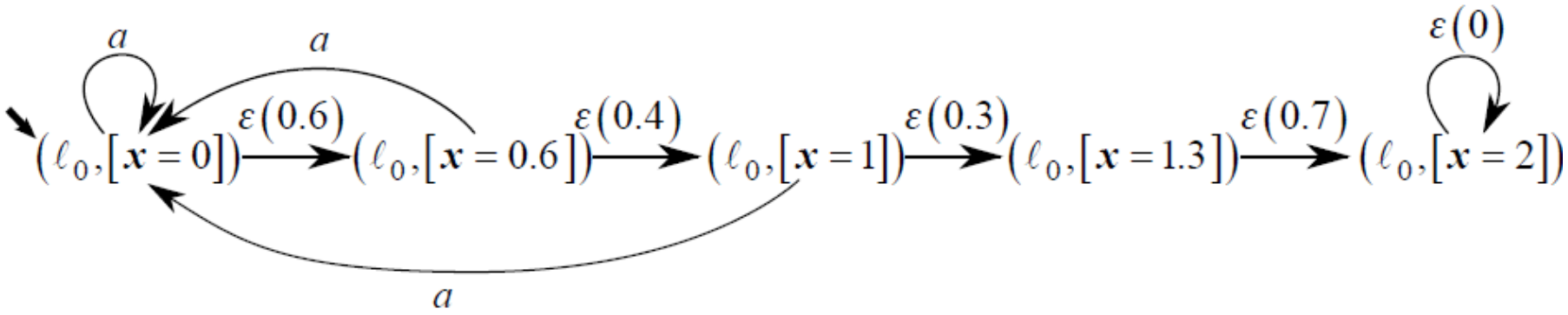
Feltesszük, hogy $v_0 \models I(\ell_0)$.

példa:



$\mathcal{A}$ időzített automata

$\mathcal{A}$-nak egy állapota van az ℓ_0, ehhez az $I(\ell_0) = x \le 2$ invariáns tartozik . Egy átmenete van, ℓ_0–ból ℓ_0-ba, melynek őrfeltétele x ≤ 1, vagyis ha ez teljesül, akkor végrehajthatja az *a* akciót és ezzel együtt az *x* óra újraindítását.



2.19. ábra. A példában szereplő $T(\mathcal{A})$ időzített átmeneti rendszer néhány átmenete

Időzített automaták hálózata

szinkronizáció: csatornák (szinkronizációs) feletti kommunikációs átmenetekkel;

$Ch = \{a, b, \ldots\}$ csatornák halmaza, $Ch \cap C = Ch \cap A = \emptyset$;

kommunikációs átmenetek címkéi egy-egy csatornához kötődnek, párokban fordulnak elő;

jelölés:

$\forall a \in Ch$ esetén $a!$ és $a?$ az a csatornához tartozó kommunikációs átmenet pár;

$a!$: a komponens szinkronizációs igényt jelez,

$a?$: a komponens kommunikációs igényt fogad;

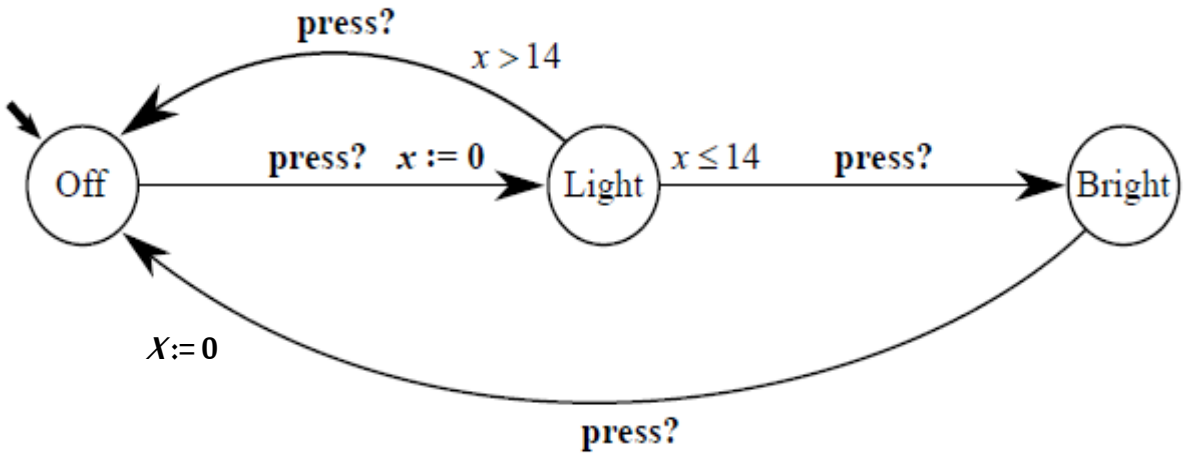
kommunikáció megvalósulás: a két komponens egyszerre képes a küldő és a fogadó kérést megvalósítani (kézfogás típusú kommunikáció);

kommunikációs átmenetek eredeti címkéi a hálózat működését reprezentáló időzített átmeneti rendszerben nem jelennek meg, helyettük a szinkronizációt a τ speciális jel jelöli az átmenetnél;

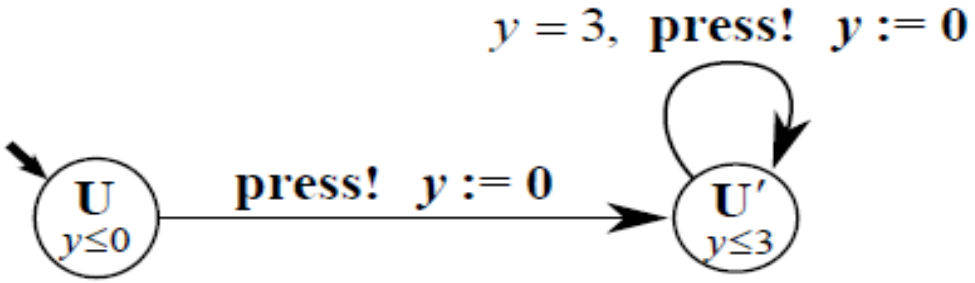
kommunikációs átmenet nem igényel időt (adatátadás ilyenkor nincs);

példa:
Hálózatnak komponense az érintőkapcsolóval ellátott lámpa és egy felhasználó, aki minden 3. időegység leteltekor megnyomja a kapcsolót.
A kommunikáció megvalósul a *press* csatornán keresztül a *press*! és *press*? címkéjű átmenetek szinkronban való végrehajtásával.

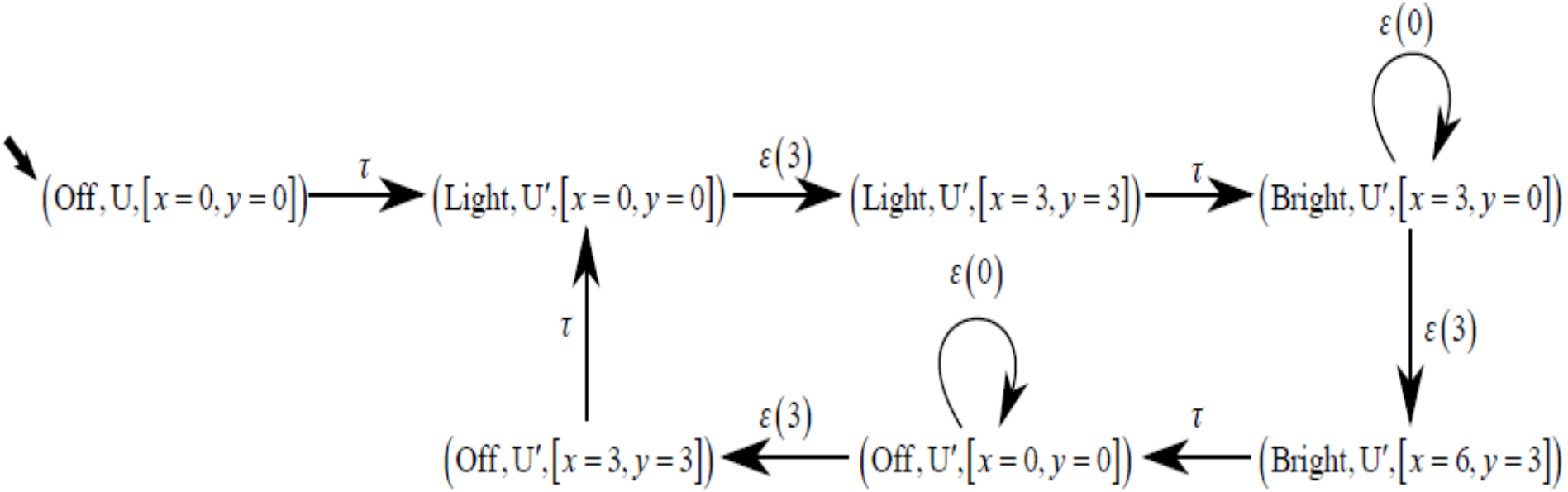
Érintőkapcsolós lámpa időzített automatája ($\mathscr{A}_1$):



Felhasználót modellező időzített automata ($\mathscr{A}_2$):



$(\mathscr{A}_1, \mathscr{A}_2)$ rendszer működését modellező időzített átmeneti rendszer néhány átmenete:



$\mathcal{A}_i = (L_i, \ell_0^i, E_i, I_i)$ $1 \le i \le m$-re egy-egy időzített automata a C óra-, Ch csatorna-, A akcióhalmazok felett;
$A = A_c \cup A_n$, ahol $A_c = \{a!, a? \mid a \in Ch\}$ a kommunikációs, A_n a normál akciók halmaza és $A_c \cap A_n = \emptyset$;

$\mathcal{A}_i$, $1 \le i \le m$ időzített automaták hálózatát $\langle \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m \rangle$ jelöli, ami az automaták szorzata;

$\mathcal{A}$-hoz rendelt $T(\mathcal{A})$ időzített átmeneti rendszer (az $\mathcal{A}$ szemantikáját vagyis működését definiálja) az alábbi címkézett átmeneti rendszer:
címkehalmaza: $A_{pr} = A_n \cup \{\tau\} \cup \{\varepsilon(d) \mid d \in \mathbb{R}_{\ge 0}\}$;
állapotai:

$$S = \{(\ell_1, \ell_2, \ldots, \ell_m, v) \mid \ell_i \in L_i,\ 1 \le i \le m,\ v: C \to \mathbb{R}_{\ge 0},\ v \models \wedge_{i \in \{1,\ldots,m\}} I_i(\ell_i)\};$$

átmenetei:

(1) $(\ell_1, \ldots, \ell_i, \ldots, \ell_m, v) \xrightarrow{a} (\ell_1, \ldots, \ell_i', \ldots, \ell_m, v')$, ahol $a \in A_n$,
ha $\exists(\ell_i \xrightarrow{g,a,R} \ell_i') \in E_i$, $v \models g$, $v' = v[R \to 0]$,
$v' \models I_i(\ell_i') \wedge \bigwedge_{k \neq i} I_k(\ell_k)$;

(2) $(\ell_1, \ldots, \ell_i, \ldots, \ell_j, \ldots, \ell_m, v) \xrightarrow{\tau} (\ell_1, \ldots, \ell_i', \ldots, \ell_j', \ldots, \ell_m, v')$, $i \neq j$,
ha $\exists(\ell_i \xrightarrow{g_i,\alpha,R_i} \ell_i') \in E_i$, $(\ell_j \xrightarrow{g_j,\beta,R_j} \ell_j') \in E_j$, $d \in Ch$, hogy
$\{\alpha, \beta\} = \{d!, d?\}$, $v \models g_i \wedge g_j$, $v' = v[(R_i \cup R_j) \to 0]$,
$v' \models I_i(\ell_i') \wedge I_j(\ell_j') \wedge \bigwedge_{k \neq i,\, j} I_k(\ell_k)$;

(3) $(\ell_1, \ell_2, \ldots, \ell_m, v) \xrightarrow{\varepsilon(d)} (\ell_1, \ell_2, \ldots, \ell_m, v+d)$, $d \in \mathbb{R}_{\geq 0}$-ra, ha
$v+d' \models \bigwedge_{i \in \{1, \ldots, m\}} I_i(\ell_i)$, $\forall\, d' \in [0, d]$-re;

kezdőállapot: $(\ell_0^1, \ell_0^2, \ldots, \ell_0^n, v_0)$, ahol $v_0(x)=0$, $\forall\, x \in C$-re.

TCTL (**T**imed **C**omputational **T**ree **L**ogic)

CTL valós idejű változata;

C órahalmaz, A véges akcióhalmaz feletti $\mathcal{A} = (L, L_0, E, I)$ időzített automaták tulajdonságai adhatók meg;

Legyen $\rho : L \to \mathcal{P}(AP)$, a helyek címkéző-függvénye.

jelölés:

$\mathcal{A} = (L, A, C, L_0, E, I, AP, \rho)$ a fenti időzített automatát jelöli;

$\mathcal{AB}(C)$ jelöli az atomi órafeltételek ($\wedge$ művelet nem fordul elő bennük) halmazát;

$V(C)$ a $v: C \to \mathbb{R}_{\geq 0}$ óraértékelések halmaza;

A logika állapotformulái:

- $\underline{1}$, a $\forall a \in AP$-re,
- g, $\forall g \in \mathcal{AB}(C)$-re,
- $f_1 \wedge f_2$, $\neg f$, ha f_1, f_2, f állapotformulák,
- $\mathbf{EU}_J(f_1, f_2)$, $\mathbf{AU}_J(f_1, f_2)$, ahol $J \subseteq \mathbb{R}_{\geq 0}$ természetes szám korlátú intervallum, illetve lehet jobbról végtelen is ([n, m], [n, m), (n, m], (n,m), n, m$\in\mathbb{N}$, illetve m=∞ is lehet)

További kifejezhető operátorok:

$\mathbf{EF}_J f = \mathbf{EU}_J(\underline{1}, f)$, $\mathbf{EG}_J f = \neg\mathbf{AF}_J \neg f$,

$\mathbf{AF}_J f = \mathbf{AU}_J(\underline{1}, f)$, $\mathbf{AG}_J f = \neg\mathbf{EF}_J \neg f$.

(**X** operátor TCTL-ben nincs)

jelölés:

$\pi = q_0 \xrightarrow{\tau_0} q_1 \xrightarrow{\tau_1} q_2 \longrightarrow \ldots$ végtelen út $T(\mathscr{A})$-ban,

$\tau_i \in A \cup \{\varepsilon(d) \mid d \in \mathbb{R}_{\geq 0}\}$,

$ext(\pi) = \Sigma_{i=0,1,\ldots} ext(\tau_i)$, ahol $ext(\tau_i) = \begin{cases} 0, \text{ ha } \tau_i \in A \\ d, \text{ ha } \tau_i = \varepsilon(d),\ d \in \mathbb{R}_{\geq 0}. \end{cases}$

π végtelen út idő-divergens, ha $ext(\pi) = \infty$, egyébként idő-konvergens.

$$\pi = q_0 \xrightarrow{\varepsilon(d_0^1)} \ldots \xrightarrow{\varepsilon(d_0^{k_0})} q_0 + d_0 \xrightarrow{a_0} q_1 \xrightarrow{\varepsilon(d_1^1)} \ldots \xrightarrow{\varepsilon(d_1^{k_1})} q_1 + d_1 \xrightarrow{a_1} q_2 \rightarrow \ldots$$

olyan idő-divergens végtelen út, melyben végtelen sok A-beli akció van, $d_i = \Sigma_{j=0,\ldots,k_i} d_i^j$, $\forall$ i $\geq$ 0-ra.

π út jelölése ekkor $q_0 \overset{d_0}{\Longrightarrow} q_1 \overset{d_1}{\Longrightarrow} q_2 \overset{d_2}{\Longrightarrow} \ldots$

π idő-divergens út jelölése, ha véges sok *A*-beli akciót tartalmaz

$$q_0 \overset{d_0}{\Longrightarrow} q_1 \overset{d_1}{\Longrightarrow} \ldots \overset{d_{n-1}}{\Longrightarrow} q_n \overset{1}{\Longrightarrow} q_{n+1} \overset{1}{\Longrightarrow} q_{n+2} \overset{1}{\Longrightarrow} \ldots$$

q konfiguráció (állapot) elérhető $T(\mathcal{A})$-ban, ha van kezdőkonfigurációból induló *q*-ba vezető véges átmenetsorozat.

$\mathcal{A}$ időzített automata idő-divergens, ha minden $T(\mathcal{A})$-beli elérhető konfigurációból indul idő-divergens átmenetsorozat.

Feltesszük ezután az automatáról, hogy idő-divergens.

jelölés*:*

$q=(\ell, v)$ $T(\mathcal{A})$-beli konfiguráció, $d \in \mathbb{R}_{\geq 0}$ esetén az $(\ell, v+d)$ konfigurációt $q+d$ jelöli.

TCTL formulák szemantikája:

Legyen $\mathcal{A} = (L, A, C, L_0, E, I, AP, \rho)$ véges időzített automata, $q=(\ell, v)$ $T(\mathcal{A})$-beli konfiguráció, $J \subseteq \mathbb{R}_{\geq 0}$, f egy TCTL formula.

$T(\mathcal{A}),q \models f$ reláció definiálása:

- $T(\mathcal{A}),q \models \underline{1}$ bármely $T(\mathcal{A})$-beli konfigurációra
- $a \in AP$-re $T(\mathcal{A}),q \models a \Leftrightarrow a \in \rho(\ell)$
- $g \in \mathcal{AB}(C)$-re $T(\mathcal{A}),q \models g \Leftrightarrow v \models g$
- $T(\mathcal{A}),q \models f_1 \wedge f_2 \Leftrightarrow T(\mathcal{A}),q \models f_1$ és $T(\mathcal{A}),q \models f_2$
- $T(\mathcal{A}),q \models \neg f \Leftrightarrow T(\mathcal{A}),q \not\models f$

- $T(\mathscr{A}),q \models \mathbf{EU}_J(f_1, f_2) \Leftrightarrow$ $\exists \pi = q_0 \xRightarrow{d_0} q_1 \xRightarrow{d_1} \ldots$, $q = q_0$ idő-divergens út, melyhez

  (1) $\exists i \geq 0$, $d \in [0, d_i]$, hogy $(\Sigma_{k=0,\ldots,i-1} d_k)+d \in J$, $T(\mathscr{A}),q_i+d \models f_2$ és

  (2) $\forall j \leq i$, $d' \in [0, d_j]$, ahol $(\Sigma_{k=0,\ldots,j-1} d_k)+d' < (\Sigma_{k=0,\ldots,i-1} d_k)+d$ teljesül $T(\mathscr{A}),q_j+d' \models f_1 \vee f_2$

- $T(\mathscr{A}),q \models \mathbf{AU}_J(f_1, f_2) \Leftrightarrow \forall \pi$ q-ból induló idő-divergens útra teljesül az (1) és (2).

jelölés:

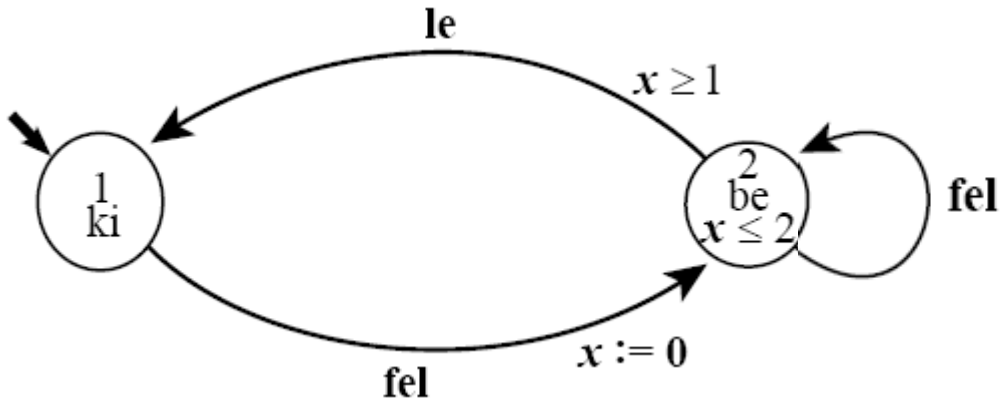
$S_f = \{q \in L \times V(C) \mid T(\mathscr{A}),q \models f\}$.

$T(\mathscr{A}) \models f \Leftrightarrow \forall \ell_0 \in L_0$-ra $(\ell_0, v_0) \in S_f$, $v_0(x) = 0$, $\forall x \in C$-re;

$\mathscr{A} \models f \Leftrightarrow T(\mathscr{A}) \models f$;

példa:

$\mathcal{A}$:



AP={ki, be},
C={*x*},
A={le, fel}

- $f_1 = \mathbf{AF}_{<1}\text{ki}$, akkor $S_{f_1} = \{\langle 1, t\rangle \mid t \geq 0\} \cup \{\langle 2, t\rangle \mid 1 < t \leq 2\}$,
- $f_2 = \mathbf{EF}_{<1}\text{ki}$, akkor $S_{f_2} = \{\langle 1, t\rangle \mid t \geq 0\} \cup \{\langle 2, t\rangle \mid 0 < t \leq 2\}$,
- $f_3 = \mathbf{AF}(\text{be} \wedge (x = 0))$, akkor $S_{f_3} = \{\langle 2,0\rangle\}$,
- $f_4 = \mathbf{AF}(\text{be} \wedge (x = 1))$, akkor $S_{f_4} = \{\langle 2, t\rangle \mid 0 \leq t \leq 1\}$.

$$\mathcal{A} \models f_1 \text{ és } \mathcal{A} \models f_2.$$

Petri háló

Petri háló (PN) egy (P, T, E, w, m_0) rendszer, ahol

P a *helyek* véges halmaza;

T a *tranzíciók* véges halmaza ($P \cap T = \emptyset$);

$E \subseteq (P \times T) \cup (T \times P)$ az *élek* véges halmaza;

w: $E \rightarrow \mathbb{N}^+$ a *súlyfüggvény;*

token eloszlás (vagy PN állapota): m: $P \rightarrow \mathbb{N}$ függvény, mely adott állapotban megadja a helyeken levő tokenek számát;

m_0 a *kezdeti token eloszlás* (a hálózat kezdeti állapota);

Petri háló struktúra (P, T, E, w).

Gráf reprezentáció

PN irányított, súlyozott, páros gráf;

hely: körökkel reprezentált csúcs;

tranzíció: rövid vízszintes vonallal (vagy téglalap) reprezentált csúcs;

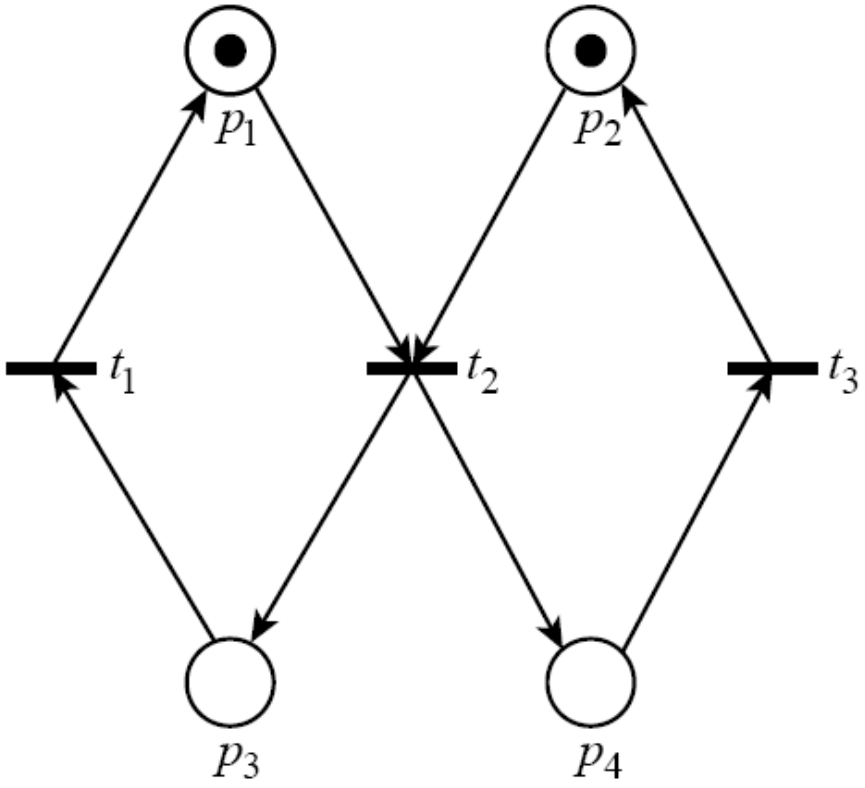
$(p, t) \in (P \text{x} T)$: p-ből t-be irányított él;

$(t, p) \in (T \text{x} P)$: t-ből p-be irányított él;

$e \in E$ él címkéje $w(e)$; (1 értéket él címkeként nem szerepeltetjük általában);

m: $P \rightarrow$N token eloszlás: $m(p)$-t a p helyen $m(p)$ darab korong reprezentálja az m állapotban;

Petri hálót egyszerűnek nevezünk, ha az élek súlya 1.
Az alábbi Petri háló egyszerű Petri hálóra példa:



2.11. ábra. Példa Petri hálóra

jelölés:

Pre: $T \rightarrow \mathcal{P}(P)$, ahol $Pre(t) = \{p \in P \mid (p, t) \in E\}$ (*t* őseinek vagy bemenő helyeinek halmaza);

Post: $T \rightarrow \mathcal{P}(P)$, ahol $Post\ (t) = \{p \in P \mid (t, p) \in E\}$ (*t* utódainak vagy kimenő helyeinek halmaza);

$\forall p \in P$, $\forall t \in T$ esetén

$w^-((p, t)) = w((p, t))$ ha $(p, t) \in E$, különben $w^-((p, t)) = 0$;

$w^+((t, p)) = w((t, p))$ ha $(t, p) \in E$, különben $w^+((t, p)) = 0$;

W súlyozott szomszédsági $|T| \times |P|$ dimenziós mátrix, ahol $W(t, p) = w^+((t, p)) - w^-((p, t))$;

W^- és W^+ szintén $|T| \times |P|$ dimenziós mátrixok, ahol

$W^-(t, p) = w^-((p, t))$, $W^+(t, p) = w^+((t, p))$.

m token eloszlásban (állapotban) egy *t* tranzíció tüzelés megengedett, ha *t* minden ősén van legalább $w((p,t))$ token, vagyis

$$\forall p \in Pre(t) : m(p) \geq w((p,t)) \; .$$

- *m* állapotban (token eloszlás) véletlen módon választ egy engedélyezett tranzíciót, melyet tüzel;
- a tüzelés $[0, \infty)$ időintervallumban bármikor megtörténhet;
- *m* állapotban a *t* tranzíció tüzelésének eredménye *m'* állapot lesz, ahol $\forall p \in P$ -re: $m'(p) = m(p) - w^{-}((p, t)) + w^{+}((t, p))$;

- m token eloszlás tekinthető egy $|P|$ elemű M oszlopvektornak, ahol az M elemei a P-beli helyeken az m szerinti tokenek számai;

- e_t jelölje a t tranzíciónak megfelelő oszlop egységvektort;

- ha a t tranzíció tüzelése történik az M állapotban, akkor ennek eredménye M' új állapot,

  ahol $M' = M + W^T \cdot e_t$

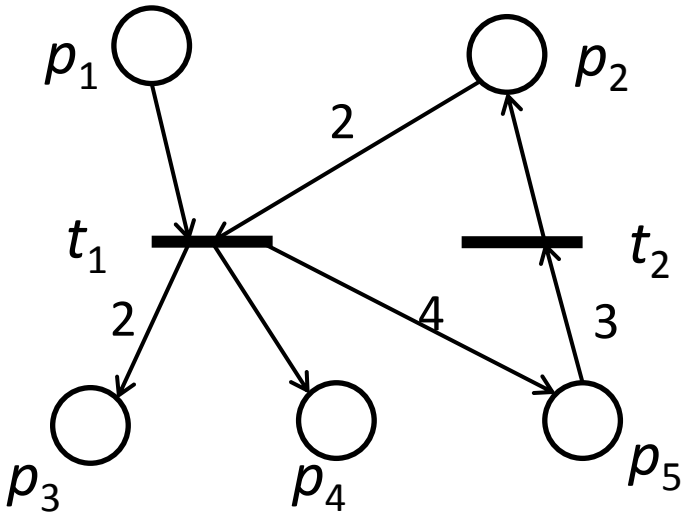
- M' állapot elérhető az M állapotból, ha megadható olyan $M = M_{i_0}, \ldots, M_{i_n} = M'$ állapot sorozat és $t_1, \ldots, t_n$ tranzíció sorozat, hogy minden $j \in \{1, \ldots, n\}$-re M_{i_j} az $M_{i_{j-1}}$-ből a t_j tranzíció tüzelésével áll elő;

- ha $<M_{i0}\ t_1 \ldots t_n\ M_{in}>$, akkor $\underline{v} = < t_1 \ldots t_n >$ tüzelési szekvencia;
- M' állapot elérhető az M állapotból a $\underline{v}$ tüzelési szekvencia által jelölése $M \rightarrow \underline{v} \rightarrow M'$;
- Petri háló tiszta, ha nincsenek önhurkai, vagyis $\forall t \in$ T-re $Pre(t) \cap Post(t) = \emptyset$;
- t tranzíció forrás tranzíció, ha $Pre(t) = \emptyset$;
- t tranzíció elnyelő tranzíció, ha $Post(t) = \emptyset$;
- t' és t'' tranzíciók függetlenek, ha

  $Pre(t') \cap (Pre(t'') \cup Post(t'')) = \emptyset$ és

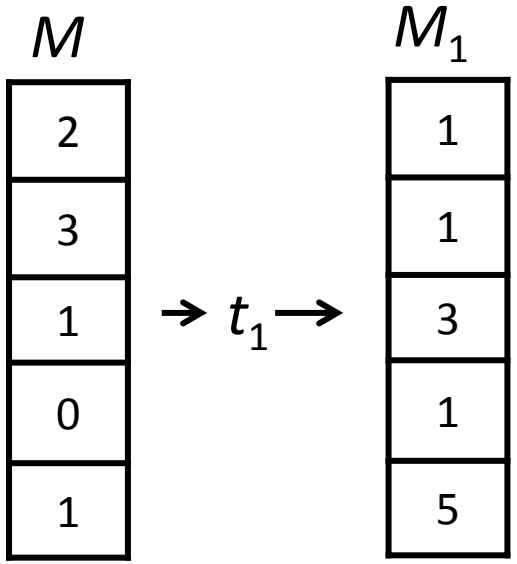
  $Pre(t'') \cap (Pre(t') \cup Post(t')) = \emptyset$;

példa szomszédsági mátrixra, tranzíció tüzelésre:



w	p_1	p_2	p_3	p_4	p_5
t_1	-1	-2	2	1	4
t_2	0	1	0	0	-3

Az alábbi M token eloszlásból t_1 tranzíció tüzelése engedélyezett és a tüzelés eredménye M_1 token eloszlás:



Petri hálóból származtatott címkézett átmeneti rendszer (LTS)

(P, T, E, w, m_0) Petri hálóból származtatott LTS rendszerben

állapotok: az m_0-ból elérhető token eloszlások;

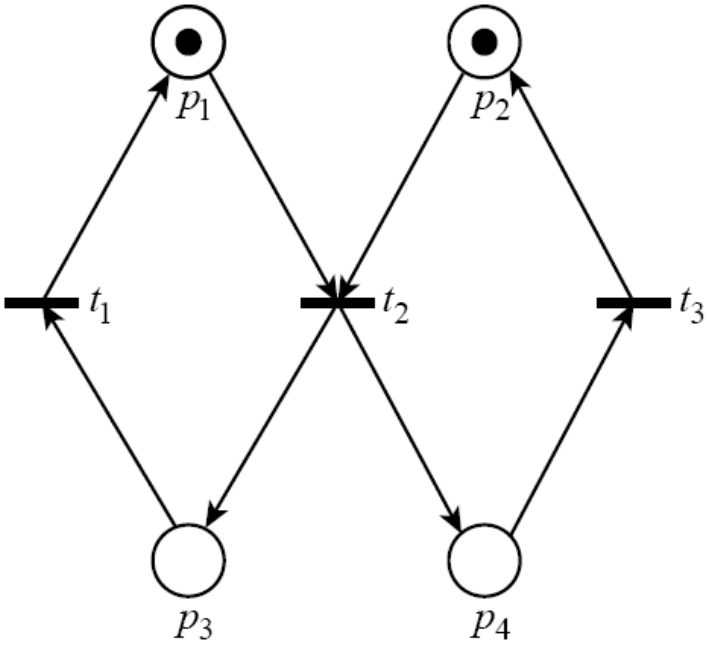
átmenetek: a tranzíciók tüzelésének felelnek meg

vagyis ha m-ben t tüzelhető és eredménye m', akkor m-ből m'-be van átmenet, melynek címkéje t, vagyis $m \mapsto t \rightarrow m'$ átmenet;

címkehalmaz: a Petri háló tranzícióinak halmaza;

kezdőállapot: m_0

példa: PN-hez LTS megadása



2.11. ábra. Példa Petri hálóra

m_0 és belőle elérhető token eloszlások:

	m_0	m_1	m_2	m_3
p_1	1	0	1	0
p_2	1	0	0	1
p_3	0	1	0	1
p_4	0	1	1	0

Legyen a példában szereplő Petri háló kezdeti token eloszlás: m_0
Átmeneti rendszer a példához:
címkehalmaz: $\{t_1, t_2, t_3\}$,
állapothalmaza: $\{m_0, m_1, m_2, m_3\}$,
átmenethalmaza:

$m_0 \mapsto t_2 \rightarrow m_1$,

$m_1 \mapsto t_1 \rightarrow m_2$,

$m_1 \mapsto t_3 \rightarrow m_3$,

$m_2 \mapsto t_3 \rightarrow m_0$,

$m_3 \mapsto t_1 \rightarrow m_0$,
kezdőállapot: m_0.

Korlátos Petri háló:

PN kiegészítve egy k: $P \rightarrow N^+$ kapacitás függvénnyel, mely minden helyhez megadja az oda elhelyezhető tokenek maximális számát.

- t tranzíció tüzelés engedélyezett m állapotban:
  $\forall p \in Pre(t)$-re $m(p) \geq w((p, t))$ és
  $\forall p \in Post(t)$-re $m(p)+W((t, p)) \leq k(p)$;
- m állapotban t engedélyezett tranzíció tüzelés eredménye, mint a végtelen kapacitású Petri hálóknál;
- korlátos PN-ből származtatott LTS véges;
- korlátos PN *kifejező ereje* nem nagyobb a végtelen kapacitásúnál vagyis megadható vele ekvivalens végtelen kapacitású PN;

Prioritásos Petri háló

Petri háló kiegészítve egy π: $T \rightarrow N^+$ prioritás függvénnyel.

- *m* állapotban az engedélyezett tranzíciók közül a legmagasabb prioritású tranzíciók valamelyikének tüzelése történik;
- prioritással kiterjesztett PN *kifejező ereje* nagyobb, a kiterjesztés nélküli PN kifejező erejénél (vagyis vannak olyan rendszerek, melyek kiterjesztés nélküli Petri hálóval nem modellezhetők);
- prioritásos Petri hálók kifejező ereje a Turing-gépekével azonos;

Állapotegyenletek

- a token eloszlások változását írják le;
- feltesszük, hogy a Petri háló tiszta;
- $\underline{v} = <M_{i0}\, t_1 \ldots t_n\, M_{in}> = < t_1 \ldots t_n>$ tüzelési szekvencia, vagyis $M_{i0} \rightarrow \underline{v} \rightarrow M_{in}$;
- ha $\underline{v}$ engedélyezett tüzelési szekvencia, akkor $\forall t_j \in \underline{v}$ átmenetre $M_{ij-1} \geq W^{-T} \cdot e_{t_j}$;
- ha $\underline{v}$ szerint történik a tranzíciók tüzelése, akkor

$$M_{ij} = M_{ij-1} - W^{-T} \cdot e_{t_j} + W^{+T} \cdot e_{t_j} = M_{ij-1} + W^{T} \cdot e_{t_j} ;$$

- token eloszlás módosulása (a fenti egyenleteket összeadva, átrendezve)

$$M_{in} - M_{i0} = W^{T} \cdot \sum_{j \in \{1,\ldots, n\}} e_{t_j} = W^{T} \cdot \underline{v}_T ,$$

ahol $\underline{v}_T$ $|T|$ elemű tüzelési szám vektor (oszlopvektor), $\underline{v}_T(t)$ a t tranzíció $\underline{v}$-ben való előfordulásainak száma;

Strukturális tulajdonságok

kezdőállapottól független tulajdonságok;

Tüzelési invariáns (*T-invariáns*)

$\underline{v}_T$ tüzelési szám vektor T-invariáns, ha

$$W^T \cdot \underline{v}_T = \underline{0} \quad \text{(oszlopvektor)}$$

lineáris egyenletrendszer megoldása;

- az egyenlet megoldások lineáris kombinációi is megoldások;
- ha $\underline{v}$ szekvencia végrehajtható M_i állapotban és T-invariáns, akkor $M_i \rightarrow \underline{v} \rightarrow M_i$ vagyis az állapottér ciklikus;
- bármely $\underline{v}$ tüzelési szekvencia az $M_0 \geq W^{-T} \cdot \underline{v}_T$ kezdőállapotból végrehajtható;

Hely invariáns (*P-invariáns*)

$\underline{r}_P$ súlyvektor (|*P*| elemű oszlopvektor) P-invariáns, ha tetszőleges tüzelési szekvenciára álladó az elért helyeken a tokenek $\underline{r}_P$ szerinti súlyozott összege, vagyis $\underline{r}_P^T \cdot M$ = állandó;

- $\forall M, M'$ állapotra, $\underline{v}$ tüzelési szekvenciára, ha $M \rightarrow \underline{v} \rightarrow M'$, akkor $M' = M + W^T \cdot \underline{v}_T \Rightarrow$
$\underline{r}_P^T \cdot M' = \underline{r}_P^T \cdot M + \underline{r}_P^T \cdot W^T \cdot \underline{v}_T \Rightarrow \underline{r}_P^T \cdot W^T \cdot \underline{v}_T = 0 \Rightarrow$

P-invariánsok az alábbi lineáris egyenletrendszer megoldásai

$$W \cdot \underline{r}_P = \underline{0};$$

Kezdőállapottól függő tulajdonságok

Elérhetőségi analízis vizsgálja, hogy m_0 kezdőállapotból egy m állapot valamely tüzelési szekvencia mellett elérhető-e vagy nem.

jelölés: legyen (N, m_0) egy PN, akkor

$$R(N, m_0) = \{m \mid \exists \underline{v}: m_0 \rightarrow \underline{v} \rightarrow m\},$$

$$L(N, m_0) = \{\underline{v} \mid \exists m: m_0 \rightarrow \underline{v} \rightarrow m\};$$

elérhetőségi probléma: $m \in R(N, m_0)$;

- elérhetőségi probléma eldönthető (általában exponenciális időben);

Élőség

(N, m_0) PN élő, ha köztes állapotoktól függetlenül $\forall t \in T$ újra tüzelhetővé válik a későbbi működés során;

- bejárási úttól független holtpontmentességet garantál;

Korlátosság

(N, m_0) PN korlátos, ha megadható egy k szám, hogy $\forall p \in P$ helyen $\forall m \in R(N, m_0)$ helyen a tokenek száma legfeljebb k;

- (N, m_0) PN korlátos $\Leftrightarrow$ $|R(N, m_0)|$ véges;

- ha egy (N, m_0) PN korlátos és elakadás mentes, akkor van olyan $\underline{v}$ tüzelési szekvencia, hogy $\underline{v}_T > \underline{0}$ T-invariáns;
- ha egy korlátos (N, m_0) PN-ben nincs $\underline{v}_T > \underline{0}$ T-invariáns, akkor minden úton elakad;
- ha egy (N, m_0) PN korlátos és élő, akkor van minden tranzíciót tartalmazó tüzelhető $\underline{v}_T > \underline{0}$ T-invariáns;
  ha nincs ilyen T-invariáns, akkor a PN nem élő;
  ha van ilyen T-invariáns egy korlátos PN-ben abból nem következik, hogy a PN élő;
- ha egy korlátos (N, m_0) PN-ben egy t tranzíció nincs benne egyetlen T-invariánsban sem, akkor csak véges sokszor tüzelhető;

Megfordíthatóság

(N, m_0) PN megfordítható, ha $\forall m \in R(N, m_0)$-re $m_0 \in R(N, m)$;

- ciklikus működre képes a hálózat;

Visszatérő állapot

(N, m_0) PN-ben $m' \in R(N, m_0)$ visszatérő állapot, ha $\forall m \in R(N, m')$-re $m' \in R(N, m)$;

- m_0-tól m'-ig tartó inicializálás után m' biztonságos, mert minden elérhető állapotból vissza lehet térni hozzá;

Fedhetőség

(N, m_0) PN m állapotát az $m' \in R(N, m_0)$ állapot lefedi, ha $\forall p \in P$: $m'(p) \geq m(p)$ vagyis $m' \geq m$;

- ha m egy t tranzíció tüzeléséhez szükséges token eloszlás, akkor t halott (soha nem tüzelhető)$\Leftrightarrow$ $\forall m' \in R(N, m_0)$ nem fedi m állapotot;

Perzisztencia

$t \in T$ perzisztens, ha engedélyezetté válva engedélyezettsége fennáll a tüzeléséig;

PN perzisztens, ha $\forall t \in T$ perzisztens;

- ha ez a tulajdonság fennáll, akkor az eredetileg párhuzamosnak szánt működések nem befolyásolják egymást;

Fairség

- korlátosan fair egy tüzelési szekvencia, ha $\forall t \in T$ korlátosan sokszor tüzelhet mielőtt egy másik tüzelne;
- globálisan fair egy tüzelési szekvencia, ha véges vagy $\forall t \in T$ végtelen sokszor fordul benne elő;
- PN korlátosan, illetve globálisan fair, ha minden tüzelési szekvencia korlátosan, illetve globálisan fair;
- ha egy PN fair, akkor a rendszerbeli párhuzamos folyamatok nem tartják fel egymást kereszthatásaik révén; mindegyik folyamat valamikor végbemegy;

Holtpont-mentesség

$\forall m \in R(N, m_0)$ az $R(N, m) \neq \emptyset$ vagyis minden elérhető állapotban van tüzelhető tranzíció;

Színezett Petri háló (CPN)

- modellezési ereje a kiterjesztett (prioritásos) színezetlen Petri hálóéval azonos;
- színezetlen Petri háló a vizsgált rendszer tulajdonságait a struktúrájával fejezi ki, így a mérete nagy lehet;
- színezett Petri háló helyei típussal rendelkeznek, a tokeneknek színe van, mely adatértéket reprezentál;
- token-eloszlás minden helyhez megadja a hely aktuális tokenhalmazát (a hely színhalmazának valamely multi-halmaza);

- tranzícióknál őrfeltételek, engedélyezési feltételként;
- éleken súlyfüggvény helyett él-kifejezés engedélyezés feltételeként, illetve a tüzelés eredményét befolyásolja;

Adattípusok:

egyszerű adattípusok:

alap típusok (pl. int, bool, real, string),
felsorolás (pl. *with* true | false),
indexelés (pl. *index* d *with* 1..5);

összetett adattípusok:

egyszerű adattípusok feletti konstrukciókkal: unió képzés, rekord szerkezet, részhalmaz képzés (megszorítással), szorzással (n-esek halmaza), listák képzése (beépített függvényekkel támogatva);

multi-halmaz:

azonos elemből több példány lehet a halmazban ($f \in S \rightarrow$ N, ahol S halmaz, akkor $n = \Sigma_{x \in S} f(x) \cdot x$);

Pl. ha A típus elemei a, b, c, akkor A feletti multi-halmaz (jelölése A_{MS}) eleme pl. $3 \cdot a + 5 \cdot c$

műveletek multi-halmazok felett: összeadás, skaláris szorzás, összehasonlítás, számosság, kivonás;

- deklarált változók, konstansok, operátorok, mellékhatás-mentes függvények alkalmazhatók;
- nyílt kifejezés, ha tartalmaz változót ;
- nyílt kifejezés kiértékelés: a változók valamely lekötése (értéket adunk nekik) szerint történik, típusa a lehetséges kiértékelések eredményeinek halmaza;
- a változók lekötésében az átmenetek függetlenek egymástól;
- egy átmenet esetén viszont azonos értékkel kötött minden adott változó mindegyik előfordulása;
- őrfeltétel tranzíciónál: multi-halmazokon értelmezett logikai értékű kifejezés;

- él-kifejezések: függvények, melyek a színhalmazokat képezik le multi-halmazokra;
- (*p*, *t*) bemeneti élen levő él-kifejezés értéke a *p* hely típusa feletti multi-halmaz;
- (*t*, *p*) kimeneti élen levő él-kifejezés értéke a *p* hely típusa feletti multihalmaz;

t tüzelés engedélyezettség feltétele:
legyen a *t* őrfeltételében és a bemeneti élein levő él-kifejezésekben előforduló változóknak olyan lekötése, mely felett az őrfeltétel teljesül (értéke true) és mindegyik bemeneti él-kifejezés értékeként előálló multi-halmaz részhalmaza az él kiindulási helye token-halmazának;

t tüzelés eredménye:

- a *t* engedélyezettséghez tartozó változó lekötések felett a kimenő élein levő él-kifejezések kiértékeléséhez a még lekötetlenül levő változóknak a színhalmazukból véletlenszerűen választ értéket és az így előálló változó lekötések felett értékeli ki a kimenő él-kifejezéseket ;
- mindegyik kimeneti hely token-halmazát bővíti a bele vezető él él-kifejezésének számított értékével;
- minden bemeneti helyén levő token-halmazt csökkenti az adott változó lekötések feletti él-kifejezés értékével;

jelölések:

- *k* kifejezésben levő változók halmaza: var(*k*)
- *k* kifejezés típusa: type(*k*)
- *S* színhalmaz feletti multi-halmazok halmaza: S_{MS}
- bool típus: B

CPN egy $(\Sigma, P, T, A, c, g, e, m_0)$ rendszer, ahol

Σ *színhalmazok* (típusok) véges halmaza;

P *helyek* véges halmaza;

T *tranzíciók* véges halmaza ($P \cap T = \emptyset$);

$A \subseteq (P \times T) \cup (T \times P)$ az *élek* véges halmaza;

c: $P \rightarrow \Sigma$ *színfüggvény*, helyenként megadja a token színek halmazát;

g a T elemeihez *őrfeltételt* rendelő függvény:
$\forall t \in T$: $\text{type}(g(t)) = \text{B} \wedge \text{type}(\text{var}(g(t))) \subseteq \Sigma$;

e az A elemeihez *él-kifejezést* rendelő függvény:
$\forall a \in A$: $\text{type}(e(a)) = c(p)_{\text{MS}} \wedge \text{type}(\text{var}(e(a))) \subseteq \Sigma$, ahol $p \in P$ az a-nak bemeneti vagy kimeneti helye;

m_0 a P elemei felett a kezdeti token eloszlást definiáló függvény (*kezdőállapot*): $\forall p \in P$: type($m_0(p)$) = $c(p)_{MS}$;

m állapot (token eloszlás): $\forall p \in P$: type($m(p)$) = $c(p)_{MS}$

CPN gráf reprezentálás a PN reprezentációhoz hasonlóan azzal a kiegészítéssel, hogy a helyeknél a színosztály, a tranzícióknál az őrfeltétel, az éleken az él-kifejezés is címkeként szerepel;

Példa egyszerű szavazási rendszerre:

Színosztályok:

$K = \{a, b, c, 0)$	a, b, c szavazó egységeket reprezentálnak;
$D = \{p, q\}$	p elutasító, q támogató szavazatot reprezentál;
$F = K \times K$	szavazásban betöltött szerepet reprezentál, (0, x) token a koordinátort reprezentálja;

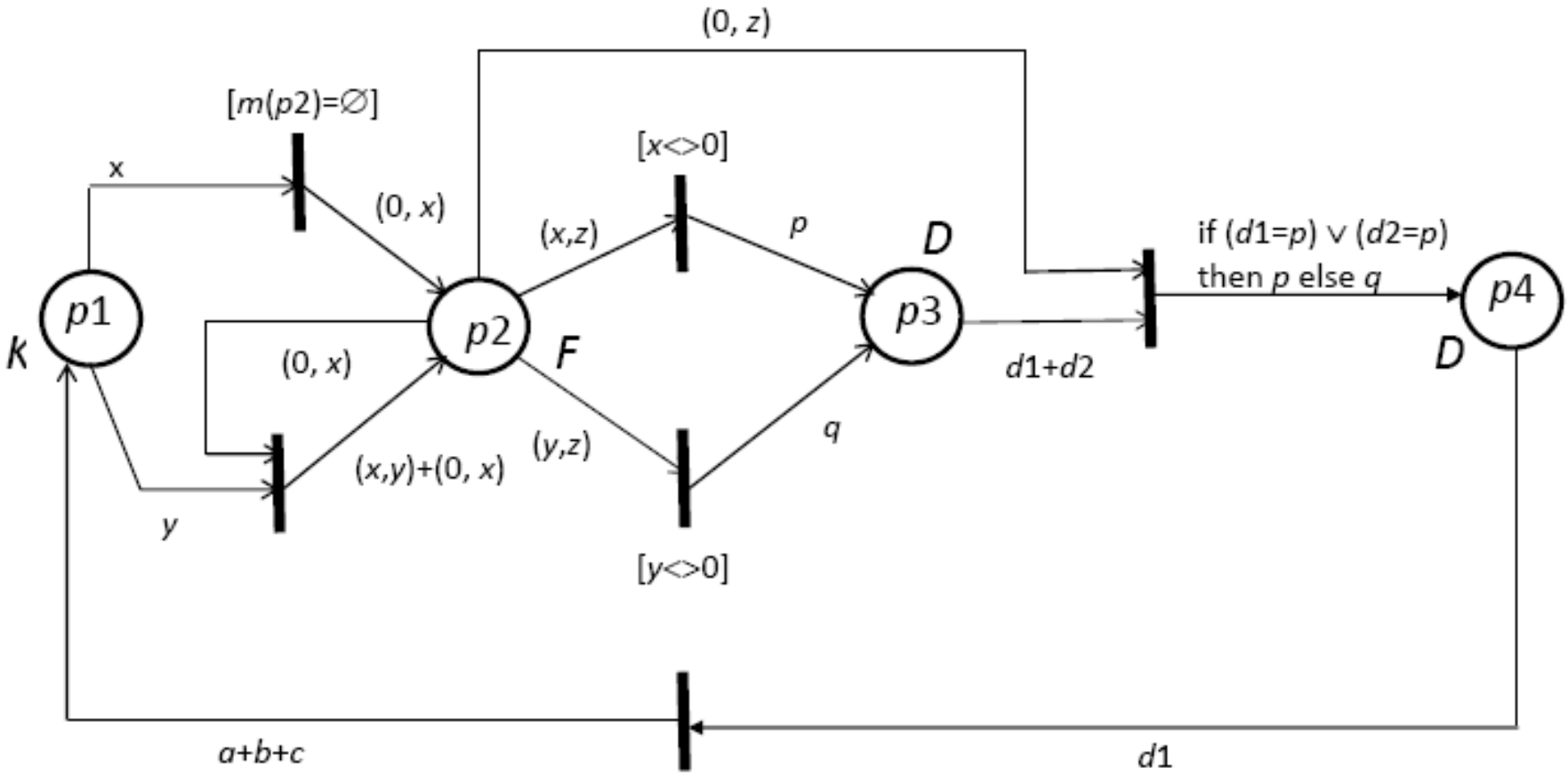
Változók:

K típusúak: x, y, z;

D típusúak: $d1$, $d2$;

Kezdeti token eloszlás (m): $m(p1) = a+b+c$, $m(p2)=m(p3)=m(p4)=\varnothing$;

Működés: egy szavazási körben 3 egység van, véletlenszerűen lesz az egyik a koordinátor; a koordinátor kéréssel fordul a többi egységhez, akik leadják a szavazatukat; a koordinátor a szavazatok alapján dönt (két támogató szavazatnál elfogadott a kérés), majd indulhat az újabb szavazási kör;



Példában adott rendszer színezett Petri hálós megvalósítása.

Processzus algebra

A nevek (akciók) nem-üres *halmaza* (input akciók);

$\overline{A} = \{\bar{a} \mid a \in A\}$ (output akciók);

τ üres akció (belső akció);

$Act = A \cup \overline{A} \cup \{\tau\}$ *akciók halmaza*;

K processzus nevek , változók halmaza;
feltesszük 0 (*NIL*) $\in K$;

$\mathcal{P}$ a *processzus kifejezések* (vagy termek, röviden processzusok) *halmaza* az *Act* és *K* felett;

processzusok akcióra képesek megváltozni, ennek megadási módja $p \mapsto a \rightarrow q$, ahol $p, q \in \mathcal{P}$, $a \in Act$;

0 semelyik akcióra nem változik;

processzus kifejezések (termek) az *Act* és *K* felett:

- $k \in K$, akkor $k \in \mathcal{P}$;
- $p \in \mathcal{P}$, $a \in Act$, akkor $a:p \in \mathcal{P}$ (*prefix* vagy akció művelet);
- $p_i \in \mathcal{P}$, I index-halmaz, akkor $\Sigma_{i \in I}\, p_i \in \mathcal{P}$ (*choice* művelet);
- $p, q \in \mathcal{P}$, akkor $p||q \in \mathcal{P}$ (*parallel* művelet)
- $p \in \mathcal{P}$, $L \subseteq A \cup \overline{A}$, akkor $p \backslash L \in \mathcal{P}$ (*restrikció* művelet);
- $p_i \in \mathcal{P}$, $x_i \in K$ $\forall i \in \{1,\ldots, n\}$-re változók, akkor p *where* $(x_1=p_1,\ldots, x_n=p_n) \in \mathcal{P}$ (*rekurzió* művelet)

- $p \in \mathcal{P}$, f: *Act* $\rightarrow$ *Act* átcímkéző függvény (vagyis $f(\bar{a}) = \overline{f(a)}$ $\forall a \in Act$-{τ}-ra, $f(\tau) = \tau$), akkor $p[f] \in \mathcal{P}$ (*átcímkéző* művelet);
  f megadható $[b_1/a_1, \ldots, b_n/a_n]$ formában, ha $f(a_i) = b_i$, i$\in$\{1, …, n\}-re;

műveletek kötési erőssége:

restrikció, átcímkézés, rekurzió (legerősebbek), utána prefix, parallel, choice műveletek (a kötés csökkenése szerint);

Strukturális operációs szemantika megadása:

Legyen $f(p_1,\ldots, p_n) \in \mathcal{P}$; $a_1,.., a_n \in Act$, ahol f egy n-változós művelet az algebrában, $p_1,\ldots, p_n \in \mathcal{P}$;

szemantikus szabályok megadásának módja:

$$\frac{p_1 \mapsto a_1 \to p_1', \ldots, p_n \mapsto a_n \to p_n'}{f(p_1,\ldots,p_n) \mapsto b \to f'(p_1,\ldots,p_n,p_1',\ldots,p_n')} \quad , C$$

ahol f és f' az algebra műveletei, C a szabály alkalmazásának feltétele (üres is lehet);

jelentése:

ha a vonal feletti átmenetek végrehajthatók és teljesülnek a C-ben megadott feltételek, akkor a vonal alatti átmenet végrehajtható;

Szemantikus szabályok:

prefix művelet:

$a : p$ processzus az a akcióra vár, melynek hatására p processzusként viselkedik vagyis

$$\frac{}{a : p \mapsto a \to p}$$

choice (alternatív választás *két tagnál*) művelet:

$$\frac{p \mapsto a \to p'}{p+q \mapsto a \to p'} \quad , \quad \frac{q \mapsto b \to q'}{p+q \mapsto b \to q'}$$

choice művelet (általánosítva *index-halmazra*):

$$\frac{p_i \mapsto a \to p_i'}{\Sigma_{i \in I}\, p_i \mapsto a \to p_i'} \quad , i \in I$$

parallel (párhuzamos kompozíció) művelet:

$$\frac{p \mapsto a \to p'}{p \,||\, q \mapsto a \to p' \,||\, q} \quad , \quad \frac{q \mapsto b \to q'}{p \,||\, q \mapsto b \to p \,||\, q'} \quad ,$$

$$\frac{p \mapsto a \to p',\ q \mapsto b \to q'}{p \,||\, q \mapsto \tau \to p' \,||\, q'} \quad , \text{ ahol } b = \bar{a}.$$

restrikció (megszorítás vagy elrejtés) művelet:

$$\frac{p \mapsto b \rightarrow p'}{p \setminus L \mapsto b \rightarrow p' \setminus L}$$, ahol $b, \bar{b} \notin L$

rekurzió művelet:

$$\frac{p\,(x_1 / p_1, \ldots, x_n / p_n) \mapsto a \rightarrow p'}{q \mapsto a \rightarrow q'}$$, ahol

$q = p$ *where* $(x_1 = p_1, \ldots, x_n = p_n)$,
$q' = p'$ *where* $(x_1 = p_1, \ldots, x_n = p_n)$,
$p(x_1/p_1, \ldots, x_n/p_n)$ jelöli azt a processzust, melyet úgy kapunk, hogy p-ben minden x_i előfordulás helyettesítve van p_i-vel;

átcímkéző művelet:

$$\frac{p \mapsto a \to p'}{p[f] \mapsto f(a) \to p'[f]}$$

*konstans*ra vonatkozó szabály:

$$\frac{p \mapsto a \to p'}{x \mapsto a \to p'}$$, ahol $x \in K$, $p \in \mathcal{P}$ és $x=p$

$p, q \in \mathcal{P}$-re azt mondjuk, hogy q a p-nek leszármazottja, ha $\exists\, a_1, \ldots, a_n \in Act$, $p_0, \ldots, p_n \in \mathcal{P}$, $n \geq 0$, hogy $\forall i \in \{1, \ldots, n\}$-re $p_{i-1} \mapsto a_i \to p_i$, $p_0 = p$, $p_n = q$.

$\forall p \in \mathcal{P}$-re p önmaga leszármazottja.

$p \in \mathcal{P}$ processzus-kifejezésből származtatott $\mathscr{A}$ címkézett átmeneti rendszer (LTS):

$\mathscr{A} = (S, T, \alpha, \beta, \lambda)$ az *Act* címkehalmaz feletti címkézett átmeneti rendszer, ahol

$S = \{q \in \mathcal{P} \mid p$-nek q leszármazottja$\}$,

$T = \{r \mid\!\rightarrow a \rightarrow q \mid r, q \in S, a \in Act$ és r az a akció hatására q processzussá alakul$\}$;

$X = \{kezdő\}$, $S_{kezdő} = \{p\}$.

példa:

$p_0 = (x \,||\, y)\backslash a$ *where* $(x = a{:}x + b{:}0,\ y = \bar{a}{:}y + c{:}0)$

feladat:

p_0 processzusból származtatható címkézett átmeneti rendszer megadása;

jelölések a példában:

$q_1 = a{:}x + b{:}0,$

$q_2 = \bar{a}{:}y + c{:}0,$

$q_0 = (q_1 \,||\, q_2)\backslash a,$

$p_1 = (0 \,||\, q_2)\backslash a$ *where* $(x{=}a{:}x{+}b{:}0,\ y{=}\bar{a}{:}y{+}c{:}0),$

$p_2 = (q_1 \,||\, 0)\backslash a$ *where* $(x{=}a{:}x{+}b{:}0,\ y{=}\bar{a}{:}y{+}c{:}0),$

$p_3 = (0 \,||\, 0)\backslash a$ *where* $(x{=}a{:}x{+}b{:}0,\ y{=}\bar{a}{:}y{+}c{:}0);$

$$\frac{q_1 \mapsto a \to x \;,\; q_2 \mapsto \bar{a} \to y}{(q_1 \,||\, q_2)\backslash a \mapsto \tau \to (x \,||\, y)\backslash a} \;,\; \frac{q_1 \mapsto b \to 0}{(q_1 \,||\, q_2)\backslash a \mapsto b \to (0 \,||\, q_2)\backslash a}$$

$$\frac{q_2 \mapsto c \to 0}{(q_1 \,||\, q_2)\backslash a \mapsto c \to (q_1 \,||\, 0)\backslash a}$$

$$\Downarrow$$

$$p_0 \mapsto \tau \to p_0,$$
$$p_0 \mapsto b \to p_1,$$
$$p_0 \mapsto c \to p_2,$$

$$(0 \,||\, q_2)\backslash a \;(x/q_1,\, y/q_2) \mapsto c \to (0 \,||\, 0)\backslash a \;\longrightarrow\; p_1 \mapsto c \to p_3$$
$$(q_1 \,||\, 0)\backslash a (x/q_1,\, y/q_2) \mapsto b \to (0 \,||\, 0)\backslash a \;\longrightarrow\; p_2 \mapsto b \to p_3$$

p_0 processzushoz az $\mathcal{A} = (S, T, \alpha, \beta, \lambda)$ címkézett átmeneti rendszer megadása:

címkehalmaz = $\{b, c, \tau\}$;

$S = \{p_0, p_1, p_2, p_3\}$, ahol

$T = \{p_0 \mapsto \tau \to p_0,$

$p_0 \mapsto b \to p_1,$

$p_0 \mapsto c \to p_2,$

$p_1 \mapsto c \to p_3,$

$p_2 \mapsto b \to p_3\}$;

$X = \{kezdő\}$;

$S_{kezdő} = \{p_0\}$.

Fordított megfeleltetés:

$\mathcal{A} = (S, T, \alpha, \beta, \lambda)$ véges címkézett átmeneti rendszer az A címkehalmaz felett;

$\forall s \in S = \{s_1, \ldots, s_n\}$ állapothoz legyen

x_s egy processzus változó,

$p_s = \Sigma_{t \in T: \alpha(t)=s} \lambda(t) : x_{\beta(t)}$ processzus;

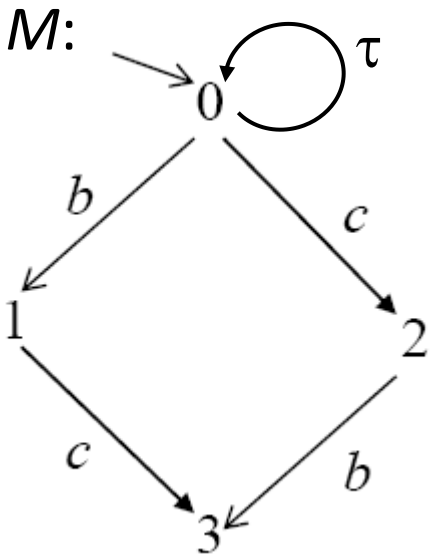
A címkehalmaz a processzus algebra *Act* akcióhalmaza.

állítás:

$\mathcal{A}$ átmeneti rendszer azon része, mely az s állapotból elérhető állapotokra megszorított a

$q_s = x_s$ *where* $(x_{s_1} = p_{s_1}, \ldots, x_{s_n} = p_{s_n})$ processzusból származtatható átmeneti rendszer.

példa: *M* átmeneti rendszerhez a q_0 processzus megadása



$p_0 = b{:}x_1 + c{:}x_2 + \tau{:}x_0$

$p_1 = c{:}x_3$

$p_2 = b{:}x_3$

$p_3 = 0$

$q_0 = x_0$ where $(x_0=p_0, x_1=p_1, x_2=p_2, x_3=p_3)$

Processzus viselkedési ekvivalenciák

Biszimuláció

R bináris reláció egy LTS állapothalmaza (S) felett biszimulációs reláció $\Leftrightarrow \forall p, q \in S$ állapotra teljesül,

ha $p\ R\ q$ és $p \mapsto a \rightarrow p'$, akkor van átmenet $q \mapsto a \rightarrow q'$

olyan, hogy $p'\ R\ q'$;

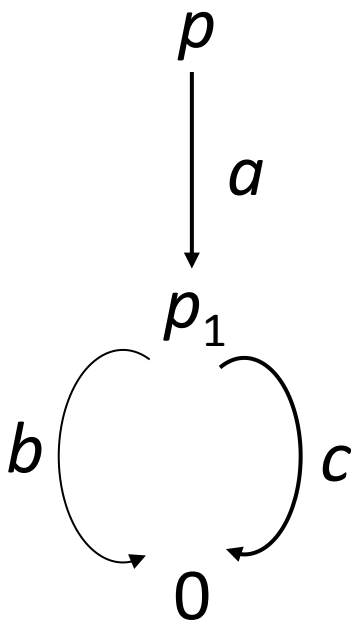
ha $p\ R\ q$ és $q \mapsto a \rightarrow q'$, akkor van átmenet $p \mapsto a \rightarrow p'$

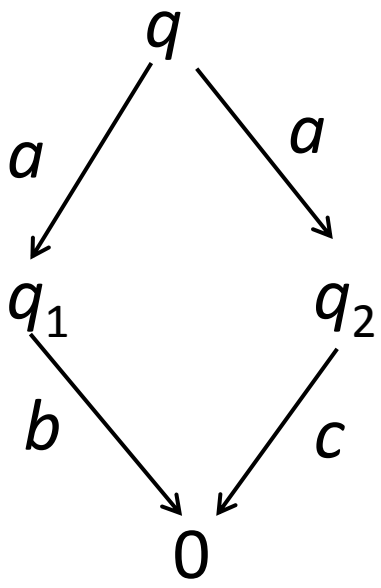
olyan, hogy $p'\ R\ q'$;

p és q állapot (processzus) biszimulár , mely kapcsolatot $p \sim q$ jelöl $\Leftrightarrow \exists R$ biszimulációs reláció, melyben $p\ R\ q$.

megjegyzés:

- $\sim$ ekvivalencia reláció;
- biszimulár processzusok ugyan azokat az akciósorozatokat hajtják végre és az elágazás struktúrájuk is hasonló.



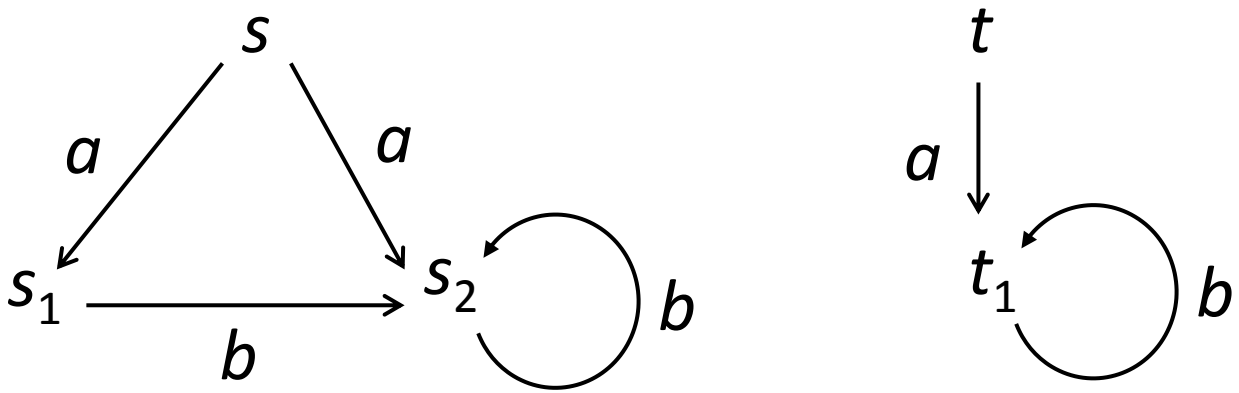


példa:

$p = a{:}(b{:}0 + c{:}0)$, $q = a{:}b{:}0 + a{:}c{:}0$

fenti LTS szerint p_1 nem lehet biszimulár q_1-el így p és q sem lehetnek biszimulárok.

példa: legyen LTS az alábbi



$R = \{(s, t), (s_1, t_1), (s_2, t_1), (s_1, s_2)\}$ biszimulációs reláció.

$(s, t) \in R \longrightarrow s \sim t$

állítás:

Legyen $p, q, r \in \mathcal{P}$. Ha $p \sim q$, akkor

- $a{:}p \sim a{:}q$ minden $a \in Act$;
- $p+r \sim q+r$ és $r+p \sim r+q$ minden $r \in \mathcal{P}$;
- $p \,||r \sim q \,||r$ és $r \,||p \sim r \,||q$ minden $r \in \mathcal{P}$;
- $p[f] \sim q[f]$ minden f átcímkézőre;
- $p\backslash L \sim q\backslash L$ minden $L \subseteq A \cup \overline{A}$.

megjegyzés:

a processzus kifejezések viselkedését nem befolyásolja, ha bennük rész-kifejezést helyettesítünk biszimulációra ekvivalens processzus kifejezéssel;

jelölés:

Legyen az $r \in \mathcal{P}$ processzusból származtatott LTS az $\mathcal{A} = (S, T, \alpha, \beta, \lambda)$ az *Act* címkehalmaz felett, $p, q \in S$, $a \in Act$.

Ha $a \neq \tau$, akkor $p \Rightarrow a \Rightarrow q \Leftrightarrow$

$$\exists\, p', q' \in S: p\ (\vdash\!\!\rightarrow \tau \rightarrow)^* p' \vdash\!\!\rightarrow a \rightarrow q'\ (\vdash\!\!\rightarrow \tau \rightarrow)^* q$$

vagy ha $a = \tau$, akkor $p\ (\vdash\!\!\rightarrow \tau \rightarrow)^* q$.

Gyenge biszimuláció

R bináris reláció az *S* felett gyenge biszimulációs reláció $\Leftrightarrow$ $\forall p, q \in S$ állapotra teljesülnek

ha $p\ R\ q$ és $p \vdash\!\!\rightarrow a \rightarrow p'$, akkor $\exists\ q \Rightarrow a \Rightarrow q'$, melyre $p'\ R\ q'$,

ha $p\ R\ q$ és $q \vdash\!\!\rightarrow a \rightarrow q'$, akkor $\exists$ p $\Rightarrow a \Rightarrow p'$, melyre $p'\ R\ q'$.

p és q állapot (processzus) gyengén biszimulár,
melyet $p \approx q$ jelöl $\Leftrightarrow \exists R$ gyenge biszimulációs reláció,
melyben $p\ R\ q$.

példa:

Legyen egy LTS az alábbi

$s \mapsto \tau \rightarrow s_1 \mapsto a \rightarrow s_2 \qquad t \mapsto a \rightarrow t_1$

Ekkor $(s, t) \notin \sim$, viszont $(s, t) \in \approx$, mivel

$R = \{(s, t), (s_1, t), (s_2, t_1)\}$ gyenge biszimuláció és

$(s, t) \in R$.

Labelled Transition System Analyzer (LTSA) verifikációs eszköz a konkurrens rendszerek számára, melynél a vizsgálandó rendszer formális modellje véges állapotú processzus (FSP) lehet. Támogatja a minimalizálást a biszimulációra és gyenge biszimulációra is.

Modell-ellenőrzés (model checking)

$\mathcal{M}$ átmeneti rendszer, *x* objektum (út, állapot vagy átmenet), *f* logikai formula (út-, állapot- vagy átmenetformula) (*x* és *f* típusa azonos)

- lokális modell-ellenőrzés: $\mathcal{M}, x \models f$ reláció vizsgálata
- globális modell-ellenőrzés: $\{x \mid \mathcal{M}, x \models f\}$ meghatározása

technikák:

- szemantika-alapú (CTL modell-ellenőrzés)
- automata-elméleti alapú (LTL modell-ellenőrzés)
- tabló-módszer alapú (HML modell-ellenőrzés)
- régió-átmeneti rendszerek konstruálása (TCTL modell-ellenőrzés)

CTL szemantika alapú modell-ellenőrzés

$\mathcal{M} = (S, T, \alpha, \beta, S_{x_1}, \ldots, S_{x_n})$ $(X, \varnothing)$-paraméteres véges átmeneti rendszer, $X=\{x_1, \ldots, x_n\}$, $\forall s \in S$ állapotból indul átmenet, f egy CTL formula;

feladat: $S_f = \{ s \in S \mid \mathcal{M}, s \models f \}$ meghatározása;

CTL modell-ellenőrzés algoritmusa

bemenet: $\mathcal{M}$ átmeneti rendszer, f CTL formula

kimenet: S_f

módszer:

- f-ről feltesszük, hogy $\{\neg, \wedge,$ **EX**, **EU**, **EG**$\}$ halmazbeli operátorokat tartalmaz (ha nem, akkor ekvivalens átalakításokat végzünk)

- S_f meghatározása az f közvetlen g részformuláihoz tartozó S_g halmazok alapján:
  - ha $f = \underline{1}$, akkor $S_f = S$,
  - ha $f = P_x$, $x \in X$, akkor $S_f = S_x$,
  - ha $f = \neg g$, akkor $S_f = S - S_g$,
  - ha $f = g_1 \wedge g_2$, akkor $S_f = S_{g_1} \cap S_{g_2}$,
  - ha $f = \mathbf{EX}g$, akkor $S_f = \{s \in S \mid \exists t \in T, \alpha(t)=s, \beta(t) \in S_g\}$,
  - ha $f = \mathbf{EU}(g_1, g_2)$, akkor S_f meghatározása:

    $S_0 := S_{g_2}$,

    $S_{i+1} := S_i \cup \{s \in S \mid s \in S_{g_1}, \exists t \in T, \alpha(t)=s, \beta(t) \in S_i\}$, $0 \leq i < |S| = m$,

    $S_f = S_m$.

    (S_f a legszűkebb olyan halmaz, mely tartalmazza S_{g_2}-t és $\forall t \in T$-re, ha $\alpha(t) \in S_{g_1}$ és $\beta(t) \in S_f$, akkor $\alpha(t) \in S_f$.)

- ha f = **EG**g, akkor S_f meghatározása:

  $S_0 := S_g$,

  $S_{i+1} := S_i \cap \{s \in S \mid \exists t \in T,\ \alpha(t)=s,\ \beta(t) \in S_i\}$, $0 \le i < |S| = m$,

  $S_f = S_m$.

  (S_f az S_g legbővebb olyan részhalmaza, melyre $\forall t \in T$-re, ha $\beta(t) \in S_f$, akkor $\alpha(t) \in S_f$.)

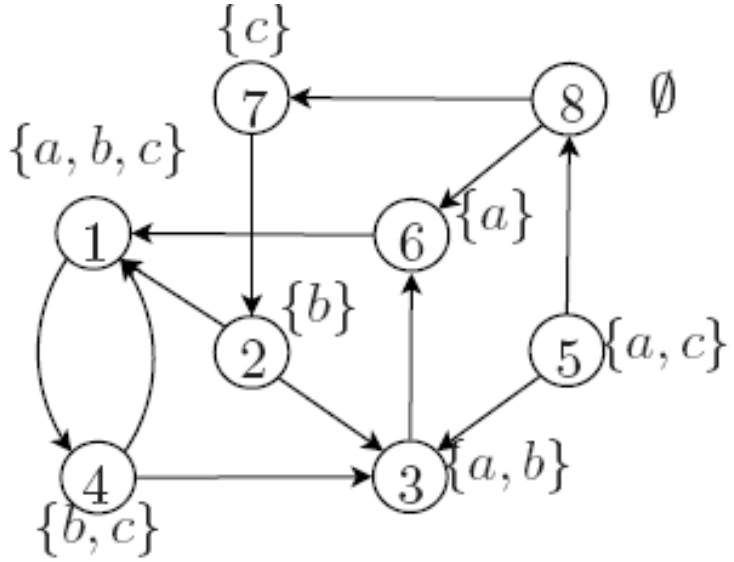
megjegyzés:

$\tau(z) : \mathcal{P}(S) \to \mathcal{P}(S)$

- f = **EU**(g_1, g_2) esetén az S_f a legkisebb fixpontja a $\tau(z) = S_{g_2} \cup [S_{g_1} \cap \{s \in S \mid \exists t \in T,\ \alpha(t)=s,\ \beta(t) \in z\}]$ leképezésnek.
- f = **EG**g esetén az S_f a legnagyobb fixpontja a $\tau(z) = S_g \cap \{s \in S \mid \exists t \in T,\ \alpha(t)=s,\ \beta(t) \in z\}$ leképezésnek.

példa:

$\mathcal{M}$:



$S_{\mathbf{EG}P_b}$ halmaz meghatározása az $\mathcal{M}$ átmeneti rendszernél:

$$S_0 = S_{P_b} = \{1, 2, 3, 4\}.$$

$$S_1 = S_0 \cap \{s \in S \mid \exists t \in T,\ \alpha(t) = s,\ \beta(t) \in S_0\} = \{1, 2, 3, 4\} \cap \{1, 2, 4, 5, 6, 7\} = \{1, 2, 4\}.$$

$$S_2 = S_1 \cap \{s \in S \mid \exists t \in T,\ \alpha(t) = s,\ \beta(t) \in S_1\} = \{1, 2, 4\} \cap \{1, 2, 4, 6, 7\} = \{1, 2, 4\}.$$

Mivel $S_1 = S_2$ így $S_{\mathbf{EG}P_b} = \{1, 2, 4\}$.

CTL modell-ellenőrző algoritmus megvalósítása *címkéző* eljárással:

- $\forall s \in S$, f CTL formula $\forall g$ részformulája esetén $s.g$ változó
- $s.g$ változó, értéke *true* lesz, ha $\mathcal{M}, s \models g$
- $s.nb$ változó, értéke az s-ből induló átmenetek száma
- *címkéző (p)*, ahol a paraméter p CTL formula, először az f-el hívódik
- *címkéző (f)* hívás eredményeként:

  $S_f = \{ s \in S \mid s.f = \text{true} \}$
- *címkéző* eljárás 7 esetre bomlik (logikai konstans, atomi kijelentés, és a megengedett alkalmazható operátorok szerint);

case$(p = \underline{1})$:

forall $s \in S$ do

$s.p := true$

endforall

case$(p = P_x)$: /* ahol $x \in \mathcal{X}$ */

forall $s \in S$ do

if $s \in S_x$ then $s.p := true$

else $s.p := false;$

endforall

case$(p = \neg g)$:

címkéző(g);

forall $s \in S$ do

$s.p :=$ not$(s.g)$;

endforall

case$(p = g_1 \wedge g_2)$:
címkéző(g_1);
címkéző(g_2);
forall $s \in S$ do
$s.p :=$ and$(s.g_1, s.g_2)$;
endforall

case$(p = \mathbf{EX}\ g)$:
címkéző(g);
forall $s \in S$ do
$s.p := false$;
endforall;
forall $t \in T$ do
if $\beta(t).g = true$ then $\alpha(t).p := true$;
endforall

case$(p = \mathbf{EU}(g_1, g_2))$:

címkéző(g_1);

címkéző(g_2);

forall $s \in S$ do

$s.p := false$;

$s.segéd := false$;

endforall

$V := \emptyset$;

forall $s \in S$ do

if $s.g_2 = true$ then

begin

$V := V \cup \{s\}$;

$s.segéd := true$;

end

endforall

```
while V≠∅ do begin
    V := V − {s};    /* kivesz V-ből egy s állapotot */
    s.p := true;
    forall t ∈ T do
        if (β(t) = s) and (α(t).segéd = false) and (α(t).g1 = true) then
        begin
            α(t).segéd := true;
            V := V ∪ {α(t)};
        end
    endforall
end
```

case$(p = \mathbf{EG}(g_1))$:

címkéző(g_1);

$V := \emptyset$;

forall $s \in S$ do

if $s.g_1 = false$ then $V := V \cup \{s\}$

else $s.p := true$;

endforall

while $V \neq \emptyset$ do begin

$V := V - \{s\}$; /* kivesz V-ből egy s állapotot */

$s.p := false$;

forall $t \in T$ do

if $(\beta(t) = s)$ and $(\alpha(t).p = true)$ then $\alpha(t).nb := \alpha(t).nb - 1$;

if $\alpha(t).nb = 0$ then $V := V \cup \{\alpha(t)\}$

endforall

end

időigény elemzés

- az eljárás terminál
- 1- 4. esetek időigény $\mathcal{O}(|S|)$
- 5-7. esetek időigény $\mathcal{O}(|S|) + \mathcal{O}(|T|))$
- teljes eljárás időigénye $\mathcal{O}(|f| \cdot (|S| + |T|))$

Ha $|S|+|T|$ az $\mathcal{M}$ átmeneti rendszer mérete, akkor a címkéző eljárás lineáris az átmeneti rendszer és a bemeneti formula méretében.

Állapotrobbanás

A rendszereket modellező átmeneti rendszer mérete a gyakorlatban nagyon nagy, a komponensek számában exponenciális méretű.

Modell-ellenőrzés hatékonyabban végrehajtható, ha az adatok explicit tárolása helyett gazdaságosabb, *redukált rendezett bináris döntési diagramok*kal (ROBDD) való reprezentáció alkalmazásával történik.

Szimbolikus modell-ellenőrzés: átmeneti rendszer megadása szimbolikusan vagyis ROBDD-kel támogatott a modell-ellenőrzés.

Halmaz reprezentálás ROBDD-vel

Bináris döntési fa (BDF):

- teljesen kiegyensúlyozott véges bináris fa
- belső csúcsok: szintenként azonos változóval címkézve, két él, egy 0 és egy 1 címkéjű indul belőlük
- különböző szinten levő címkék a fában különbözőek
- levél (terminális) csúcsok: címkéjük 0 vagy 1

t BDF-hez egyértelműen megfeleltethető egy $\varphi(x_1,\ldots, x_n)$ Boole-függvény:

- *t* gyökérétől levélig a belső csúcsok címkéi $x_1,\ldots, x_n$
- *t* gyökéréből induló utak él-címkéinek sorozatai a φ igazságtáblájának egy-egy sora, φ értéke az út végén a levél címkéje

H n-hosszú bitvektorral kódolható véges halmaz;

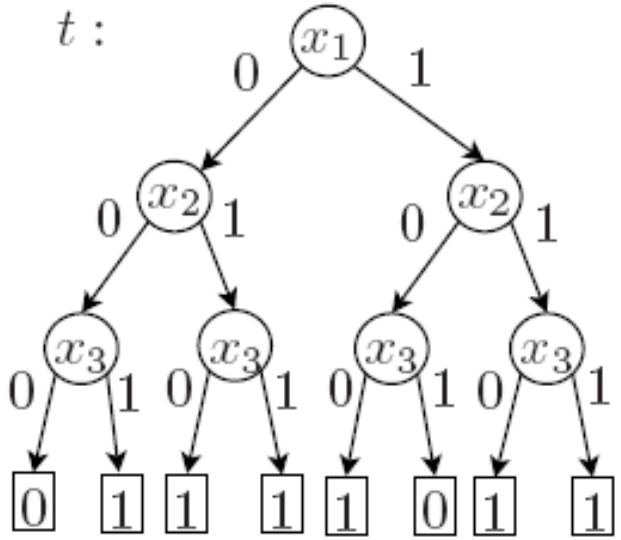
H halmazt $\varphi(x_1,\ldots, x_n)$ Boole-függvény reprezentál:

$\forall(u_1,\ldots, u_n)\in\{0, 1\}^n$-re $(u_1,\ldots, u_n)\in H \leftrightarrow \varphi(u_1,\ldots, u_n) = 1$

következmény: ha $\varphi(x_1,\ldots, x_n)$ megfelel *t* BDF-nek, akkor *t* a *H*-t reprezentálja.

példa:

H= {(0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,1,0), (1,1,1)}



x_1	x_2	x_3	φ
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Bináris döntési diagram (BDD) $X = \{x_1,\ldots, x_n\}$ felett:

- gyökércsúccsal rendelkező, ciklusmentes, véges, irányított gráf,
- belső csúcsokból két él, egy 0-val és egy 1-el címkézett él indul, címkéik X-beli változók, gyökércsúcsból elérhetők,
- levél csúcsok címkéje 0 vagy 1,
- gyökértől levélig az utak csúcsainak címkéi különbözőek;

jelölések:

- $[t]$ a t BDD gyökércsúcsa
- t_0, t_1 t-nek a $[t]$-hez 0, illetve 1 címkéjű éllel kapcsolódó rész BDD-je, ha $[t]$ belső csúcs
- $c(n)$ az n csúcs címkéje

t BDD reprezentálása $\varphi_t(x_1,\ldots, x_n)$ Boole-függvénnyel:

- ha $[t]$ levél, akkor $\varphi_t(x_1,\ldots, x_n) = c([t])$
- ha $[t]$ belső csúcs, $c([t]) = x_i$, $1 \leq i \leq n$, akkor

$\varphi_t(x_1,\ldots, x_n) = (\neg x_i \wedge \varphi_{t_0}(x_1,\ldots, x_n)) \vee (x_i \wedge \varphi_{t_1}(x_1,\ldots, x_n))$;

t_1 és t_2 BDD-ék ekvivalensek ($t_1 \equiv t_2$), akkor és csak akkor, ha $\varphi_{t_1}(x_1,\ldots, x_n) = \varphi_{t_2}(x_1,\ldots, x_n)$;

Rendezett bináris döntési diagram (OBDD) $X = \{x_1,\ldots, x_n\}$ felett :

BDD, melyhez $\exists$ X elemeinek olyan lineáris rendezése, hogy $\forall$ gyökértől levélig vezető úton a változók egymásra következése ennek nem mond ellent;

Redukált bináris döntési diagram (RBDD):

- BDD,
- nem tartalmaz izomorf rész BDD-ket,
- nincs olyan csúcsa, melynek kimenő élei ugyanabba a csúcsba vezetnek;

Minden BDD-hez megadható vele ekvivalens RBDD.

redukál algoritmus:
bemenet: *t* BDD
kimenet: *t'* RBDD ($t \equiv t'$)
módszer:

1. ha van 1 címkéjű csúcs, akkor egyet kiválasztunk, a többibe vezető éleket ebbe irányítjuk és töröljük a többi 1 címkéjű csúcsot;
   hasonlóan járunk el a 0 címkéjű csúcsokkal is;

2. ha egy *x* csúcs kimenő élei ugyanabba az *y* csúcsba vezetnek, akkor az *x*-be vezető éleket átirányítjuk az *y*-ba és töröljük az *x* csúcsot;

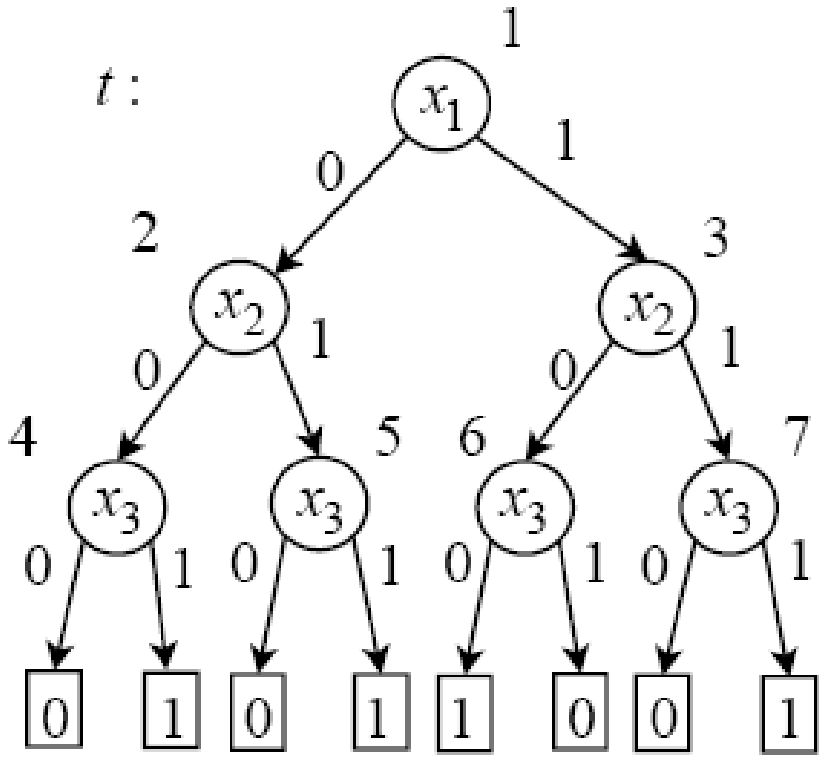
3. ha *x*, *y* belső csúcsok, $x \neq y$, $c(x) = c(y)$, *x* és *y* 0 címkéjű kimenő élei ugyanabba a csúcsba vezetnek, illetve az 1 címkéjű kimenő éleik is ugyanabba a csúcsba vezetnek, akkor pl. az *y*-ba vezető éleket átirányítjuk az *x*-be és az *y*-t töröljük a kimenő éleivel együtt;

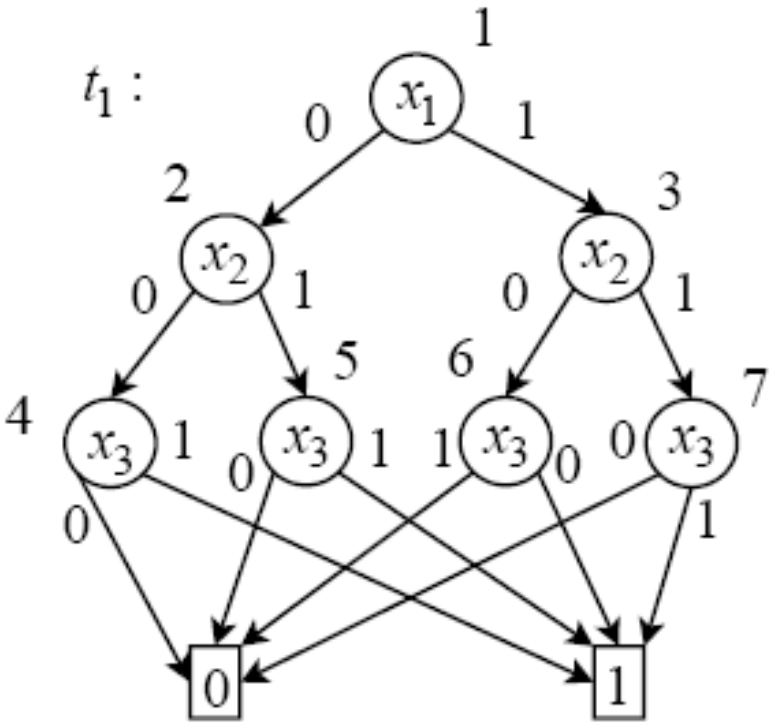
1-3 lépések ismétlődnek amíg már nem változtatják az addig kialakult BDD-t;

Redukált rendezett bináris döntési diagram (ROBDD) rendezett RBDD.

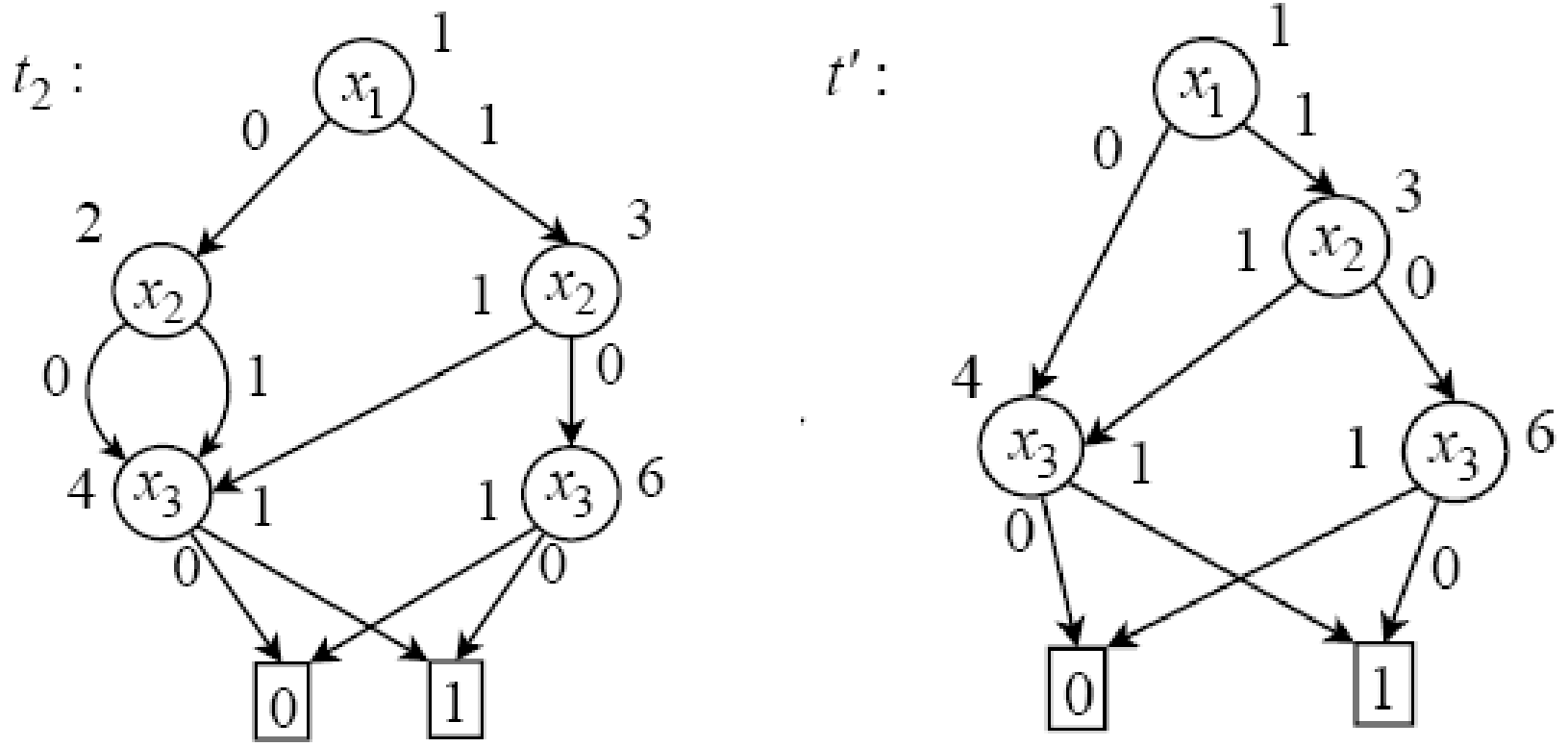
példa *redukáló* algoritmus alkalmazására

t-re az 1. lépés alkalmazásának eredménye t_1





t_1-re a 3. lépés alkalmazás eredménye t_2, erre a 2. lépés alkalmazásával áll elő a *t'* redukált BDD:



megjegyzések:

- redukáló algoritmus a változók rendezését megőrzi;
- minden BDF OBDD;
- Boole-függvény ROBDD-vel való reprezentálása függ a változók sorrendjétől. Ha f a φ, g a ψ Boole-függvényeket reprezentáló azonos változó rendezéshez tartozó ROBDD-ék, akkor $\varphi = \psi \Leftrightarrow f$ és g izomorfak.

jelölések:

- H_t a t ROBDD-vel reprezentált halmaz
- φ_H a H halmazt reprezentáló Boole-függvény
- φ_t a t ROBDD által reprezentált Boole-függvény
- $[\varphi]$ a φ Boole-függvény által reprezentált halmaz
- D_H a H halmazt reprezentáló ROBDD

Műveletek ROBDD-k felett

feltesszük: ROBDD-kben a változók sorrendje azonos

- halmaz üresség: $H_t = \varnothing \leftrightarrow c([t]) = 0$
- halmazok azonossága:
  $H_f = H_g \leftrightarrow f$ és g ROBDD-ék izomorfak
- halmaz komplementerét reprezentáló ROBDD:
  $\overline{H}_t$ halmazt reprezentáló $\bar{t}$ ROBDD a t-ből megkapható a 0 címke 1-re az 1 címke 0-ra való módosításával;
- halmazok metszetét reprezentáló ROBDD:
  $H_f \cap H_g$ halmazt (vagyis a $\varphi_f \wedge \varphi_g$ Boole-függvényt) reprezentáló t ROBDD a *redukál*(*metszet*(f, g)) eljárással konstruálható meg;

metszet (*f*, *g*):

- Ha *c*([*f*]), *c*([*g*]) változók, akkor
  - ha $c([f]) = c([g])$, akkor
    $c([t]) := c([f])$,
    $t_0 := \text{metszet}(f_0, g_0)$,
    $t_1 := \text{metszet}(f_1, g_1)$.
  - $c([f]) < c([g])$, akkor
    $c([t]) := c([f])$,
    $t_0 := \text{metszet}(f_0, g)$,
    $t_1 := \text{metszet}(f_1, g)$.
  - ha $c([f]) > c([g])$, akkor
    $c([t]) := c([g])$,
    $t_0 := \text{metszet}(f, g_0)$,
    $t_1 := \text{metszet}(f, g_1)$.

- Ha $c([f]) = 0$, akkor $t := f$.
- Ha $c([g]) = 0$, akkor $t := g$.
- Ha $c([f]) = 1$, akkor $t := g$.
- Ha $c([g]) = 1$, akkor $t := f$.

- halmazok unióját reprezentáló ROBDD:

  $H_f \cup H_g$ halmazt reprezentáló t ROBDD a *redukál*(*unió*(f, g)) eljárással konstruálható meg.

unió (f, g):

- Ha $c([f])$, $c([g])$ változók, akkor
  - ha $c([f]) = c([g])$, akkor
    $c([t]) := c([f])$,
    $t_0 := \text{unió}(f_0, g_0)$,
    $t_1 := \text{unió}(f_1, g_1)$.
  - ha $c([f]) < c([g])$, akkor
    $c([t]) := c([f])$,
    $t_0 := \text{unió}(f_0, g)$,
    $t_1 := \text{unió}(f_1, g)$.
  - ha $c([f]) > c([g])$, akkor
    $c([t]) := c([g])$,
    $t_0 := \text{unió}(f, g_0)$,
    $t_1 := \text{unió}(f, g_1)$.
- Ha $c([f]) = 0$, akkor $t := g$.
- Ha $c([g]) = 0$, akkor $t := f$.
- Ha $c([f]) = 1$, akkor $t := f$.
- Ha $c([g]) = 1$, akkor $t := g$.

- projekciót reprezentáló ROBDD:

  $t[x_i = k]$ a $\varphi_t(x_1,\ldots, x_{i-1}, k, x_{i+1},\ldots, x_n)$ Boole-függvényt reprezentáló ROBDD-t jelöli, melyet a t ROBDD $x_i = k$ *szerinti projekciójának* nevezünk, ahol $k \in \{0, 1\}$.

  $t[x = k]$ ROBDD-t megkonstruáló eljárás, ahol $x \in X$, a *redukál*(*projekció*(t, x, k));

projekció(t, x, k):

t-nek minden olyan n csúcsából induló élét, ami x-el címkézett n' csúcsba vezet, átirányítjuk az n'-ből k címkéjű éllel kapcsolt leszármazott csúcsába;

- egzisztenciális absztrakciót reprezentáló ROBDD :

$\exists x_i t$ a t ROBDD x_i változó szerinti egzisztenciális absztrakciót reprezentáló ROBDD;

$\exists x_i \varphi$ a φ Boole-függvény x_i változó szerinti egzisztenciális absztrakcióját jelöli;

$\exists x_i \varphi = \varphi(x_1,\ldots, x_{i-1}, 0, x_{i+1},\ldots, x_n) \vee \varphi(x_1,\ldots, x_{i-1}, 1, x_{i+1},\ldots, x_n)$

Ha t a φ Boole-függvényt reprezentáló ROBDD, akkor $\exists x_i t$ a $\exists x_i \varphi$ Boole-függvényt reprezentáló ROBDD-t jelöli, melyet az alábbi eljárás konstruál

redukál(*unio*($t[x_i = 0]$, $t[x_i = 1]$)), ahol

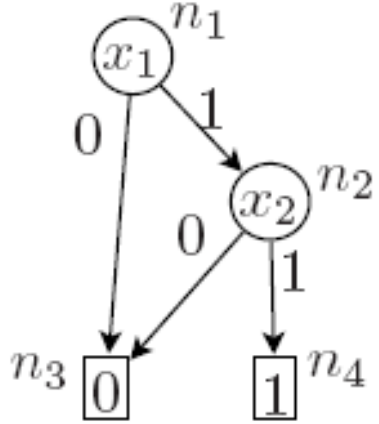
$t[x_i = 0]$ = *redukál*(*projekció*(t, x_i, 0)),

$t[x_i = 1]$ = *redukál*(*projekció*(t, x_i, 1));

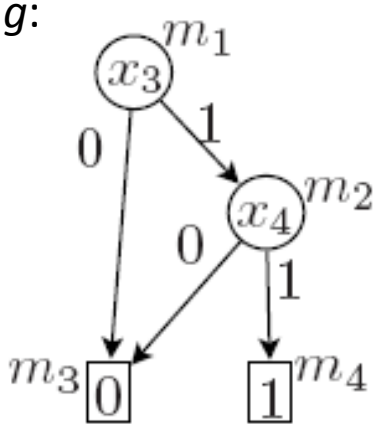
példák ROBDD-k metszetére

példa: f és g ROBDD-k ($x_1 < x_2 < x_3 < x_4$) uniójának konstruálására ($t'' = D_{H_f \cup H_g}$):

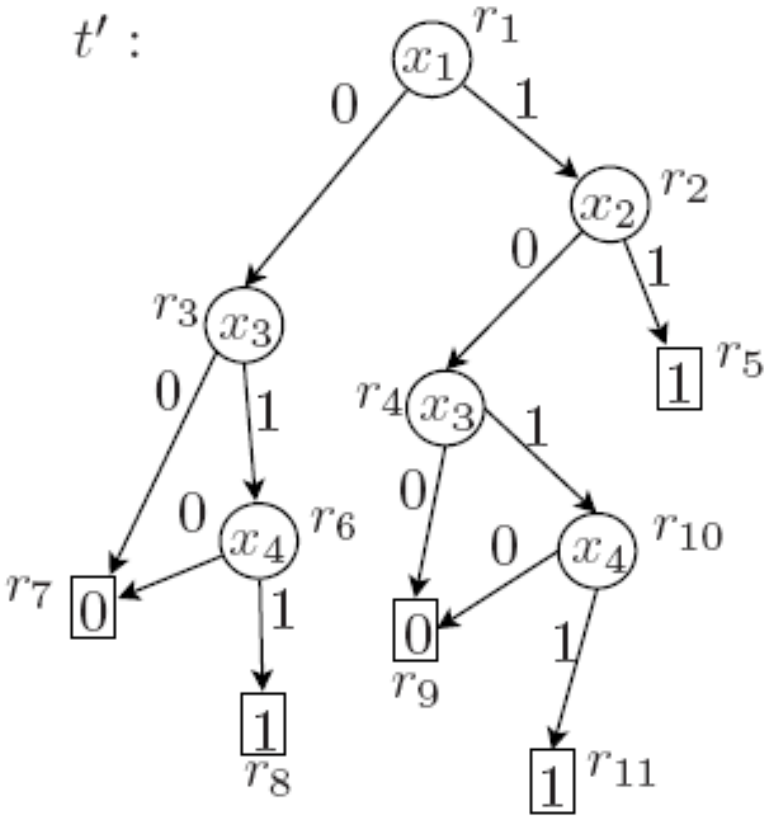
f:



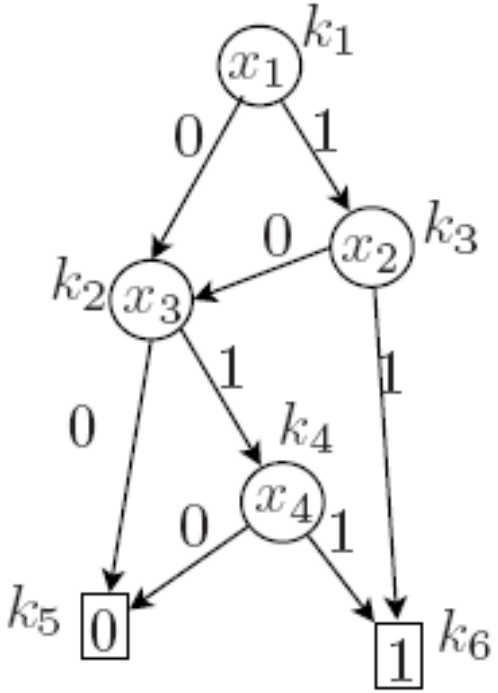
g:



t' :



t'':



Szimbolikus CTL modell-ellenőrzés

- modellben szereplő halmazok, relációk explicit tárolása helyett ezeket reprezentáló Boole-függvényeket tárolja;
- a modell-ellenőrzés algoritmusában szereplő halmazműveleteket Boole-függvények alkalmazásával végzi;
- feltesszük, hogy a Boole-függvényeket reprezentáló ROBDD-ékben a változók rendezése azonos;
- Boole-függvényeknek alkalmasan megválasztott változó rendezés melletti ROBDD-kel való tárolása jelentősen kevesebb helyet igényelhet, mint a halmazok közvetlen tárolása;

Modell reprezentálása ROBDD-ékkel

- $\mathcal{M} = (S, T, \alpha, \beta, S_{x_1},\ldots, S_{x_n})$ $(X, \varnothing)$-paraméteres véges átmeneti rendszer, $X = \{x_1,\ldots, x_n\}$, $\forall s \in S$ állapotból indul átmenet;
- $k := \lceil \log_2 (|S|) \rceil$, feltesszük S elemei k-hosszú bitvektorral kódoltak;
- $\varphi_S(z_1,\ldots, z_k)$ az S halmazt reprezentáló Boole-függvény;
- $\mathcal{M}$ átmeneteit leíró $R = \{(\alpha(t), \beta(t)) \mid t \in T\}$ halmaz reprezentálás:
  R elemeinek kódja 2k-bites, első k bit $\alpha(t)$ kódja, a további k-bit $\beta(t)$ kódja, $\forall t \in$-re;
- $\varphi_R(z_1,\ldots, z_k, z_1',\ldots, z_k')$ az R halmazt reprezentáló Boole-függvény;

- $\varphi_{S_{x_i}}(z_1,\ldots, z_k)$ az S_{x_i} halmazt reprezentáló Boole-függvény, $\forall 1 \leq i \leq n$-re;

- változók rendezése az ROBDD-ékben:

$$z_1 < z_2 < \ldots < z_k < z_1' < z_2' \ldots < z_k'$$

jelölések:

φ': φ Boole-függvényben z_i változó átnevezése z_i'-re, $\forall\ 1 \leq i \leq k$-ra;

D': D ROBDD-ben z_i változó átnevezése z_i'-re, $\forall\ 1 \leq i \leq k$-ra;

$D_{[g]}$: S_g halmazt reprezentáló ROBDD, g CTL formula;

$\bar{z}$: $z_1 \ldots z_k$ változók sorozata, $\bar{z}' = z_1' \ldots z_k'$ változók sorozata;

Szimbolikus CTL modell-ellenőrzés algoritmusa

bemenet: $\mathcal{M}$ a $\varphi_S(z_1,\ldots, z_k)$, $\varphi_R(z_1,\ldots, z_k, z_1',\ldots, z_k')$, $\varphi_{S_{x_i}}(z_1,\ldots, z_k)$, $1 \leq i \leq n$-re, Boole-függvényekkel megadva és *f* CTL formula;

kimenet: S_f halmazt reprezentáló φ_{S_f} Boole-függvény;

módszer:

- *f*-ről feltesszük, hogy {¬, ∧, **EX**, **EU**, **EG**} halmazbeli operátorokat tartalmaz (ha nem, akkor ekvivalens átalakításokat végzünk);
- *f* minden *g* közvetlen részformulájához meghatározza a φ_{S_g} Boole-függvényt, melyet a $D_{[g]}$ ROBDD reprezentál;

- ha $f = \underline{1}$, akkor $D_{[f]} = D_S$
- ha $f = \mathrm{P}_x$, $x \in X$, akkor $D_{[f]} = D_{S_x}$
- ha $f = \neg g$, akkor $D_{[f]} = \overline{\mathrm{D}}_{[g]}$
- ha $f = g_1 \wedge g_2$, akkor $D_{[f]} = \mathit{redukál}(\mathit{metszet}(D_{[g_1]}, D_{[g_2]}))$
- ha $f = \mathbf{EX}g$, akkor $D_{[f]} = \exists\, \overline{z}'(\mathit{redukál}(\mathit{metszet}(D_R, D'_{[g]})))$
- ha $f = \mathbf{EU}(g_1, g_2)$, akkor $D_{[f]}$ kiszámítása:

  $D_0 = D_{[g_2]}$,

  $D_{\mathrm{i}+1} = \mathit{redukál}(\mathit{unió}(D_{\mathrm{i}}, \mathit{redukál}(\mathit{metszet}(D_{[g_1]}, K))))$,

  $K = \exists\, \overline{z}'(\mathit{redukál}(\mathit{metszet}(D_R, D'_{\mathrm{i}})))$, $\forall 0 \leq \mathrm{i} < |S| = \mathrm{m}$

  $D_{[f]} = D_{\mathrm{m}}$

- ha f = **EG**g, akkor $D_{[f]}$ kiszámítása:

  $D_0 = D_{[g]}$,

  $D_{i+1} = redukál(metszet(D_i, K))$,

  $K = \exists \overline{z}'(redukál(metszet(D_R, D'_i)))$, $\forall\ 0 \leq i < |S| = m$

  $D_{[f]} = D_m$

Az iterációs számítások befejeződhetnek a legkisebb $0 \leq i < m$, melyre $D_{i+1} = D_i$, és ekkor $D_{[f]} = D_i$.

LTL automata-elméleti alapú modell-ellenőrzés

$\mathcal{M} = (S, T, \alpha, \beta, S_{x_1},\ldots, S_{x_n})$ $(X, \varnothing)$-paraméteres véges átmeneti rendszer, $X = \{x_1,\ldots, x_n\}$, $\forall s \in S$ állapotból van rákövetkező állapot,

$S_0 \subseteq S$ a kezdőállapotok halmaza, $AP = \{P_x \mid x \in X\}$,

f AP feletti LTL formula.

feladat: $\mathcal{M} \models f$ reláció eldöntése.

$L: S \rightarrow \mathcal{P}(AP)$: $a \in L(s) \leftrightarrow s \in S_x$, $a = P_x$ $\forall a \in AP$, $\forall s \in S$-re.

$\mathcal{M}$-beli $c = s_0,s_1,\ldots$ végtelen út lenyomata $v = v_0v_1\ldots$ végtelen $\mathcal{P}(AP)$ feletti szó, ahol $L(s_i) = v_i$, $\forall$ i $\geq$ 0-ra.

Jelölések:

$\Sigma^{\omega} = \{v_0 v_1 \ldots \mid \forall\, i \geq 0\text{-ra } v_i \in \Sigma\}$

$v^i = v_i v_{i+1} \ldots \; \forall\, i \geq 0$-ra, ha $v = v_0 v_1 \ldots$

$\mathcal{B} = (Q, \Sigma, \delta, I, F)$ (kiterjesztett) Büchi-automata:

- Q véges halmaz, *állapotok* halmaza,
- Σ véges halmaz, *input ábécé*,
- $\delta \subseteq Q \times \Sigma \times Q$ *átmenetek* halmaza,
- $I \subseteq Q$ *kezdőállapotok* halmaza,
- $F = \{F_1, \ldots, F_k\}$, $\forall\, 1 \leq i \leq k$-ra $F_i \subseteq Q$ (vagy $F = \emptyset$).

$v = v_0 v_1 \ldots \in \Sigma^{\omega}$ szónak a $\rho = q_0, q_1, \ldots$ ($q_i \in Q$, $\forall\, i \geq 0$-ra) $\mathcal{B}$-ben számítási sorozata, ha $\forall\, i \geq 0$-ra $(q_i, v_i, q_{i+1}) \in \delta$.

$inf(\rho) = \{q \in Q \mid$ végtelen sok i-re $q_i = q\}$

$v \in \Sigma^\omega$ szót elfogadja a $\mathcal{B}$ (kiterjesztett)Büchi-automata:

ha $\exists\ \rho = q_0, q_1, \ldots$ számítási sorozat a $\mathcal{B}$-ben, hogy $q_0 \in I$,

$\forall\ 1 \le i \le k$-ra $inf(\rho) \cap F_i \neq \emptyset$.

$\mathcal{B}$ Büchi-automata által felismert nyelv

$L^\omega(\mathcal{B}) = \{v \in \Sigma^\omega \mid v$-t a $\mathcal{B}$ Büchi-automata elfogadja$\}$.

Ismert állítások:

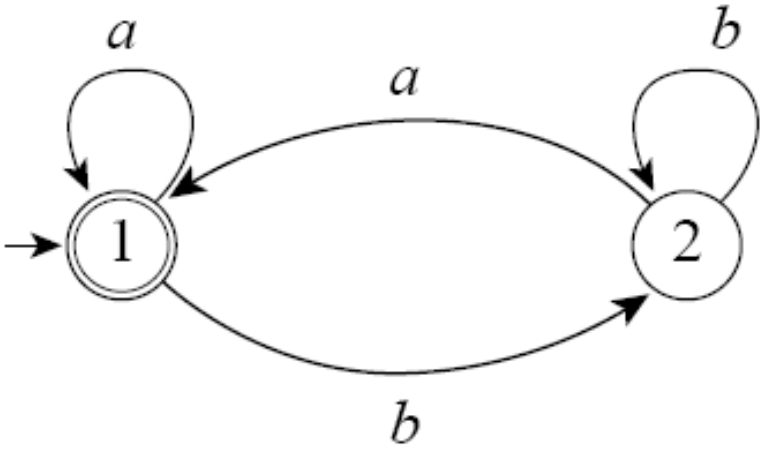
- A Büchi-automatákkal felismerhető nyelvek osztálya zárt a metszet és a komplementer képzésre.
- Tetszőleges $\mathcal{B}$ Büchi-automatára eldönthető, hogy $L^\omega(\mathcal{B}) = \emptyset$.

példa Büchi-automatára:

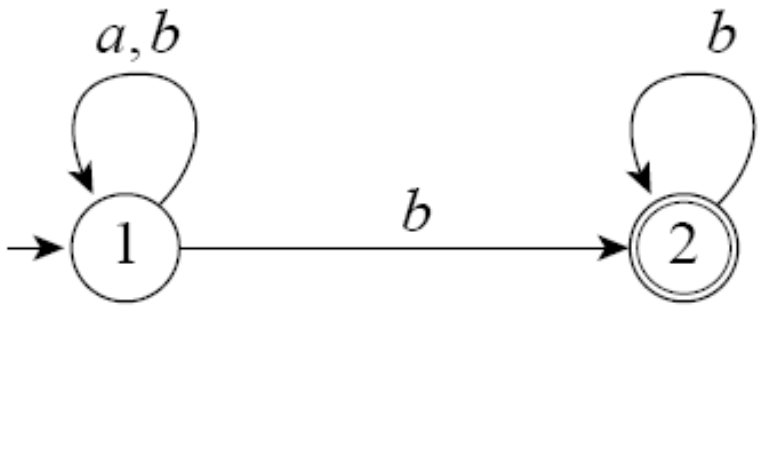
M_1 és M_2-ben F_1 elemeit kettős kör jelöli.

$L^{\omega}(M_1)= \{v \in \{a, b\}^{\omega} \mid v\text{-ben végtelen sok } a \text{ fordul elő}\}$,

$L^{\omega}(M_2)= \{v \in \{a, b\}^{\omega} \mid v\text{-ben véges sok } a \text{ fordul elő}\}$.



M_1



M_2

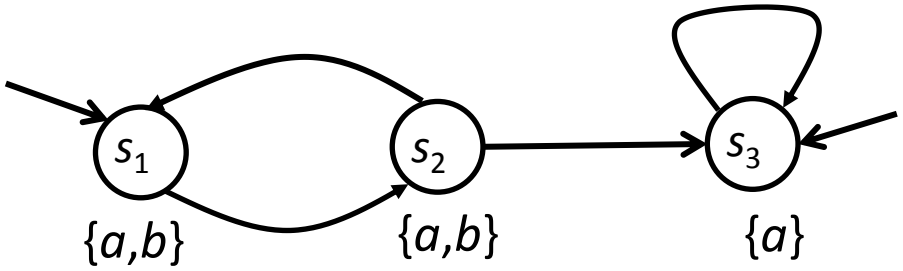
LTL modell-ellenőrzés lépései:

1. $\mathcal{M}$ átmeneti rendszerhez olyan $\mathcal{B}_{\mathcal{M}} = (Q, \Sigma, \delta, I, F)$ Büchi-automata megadása, melyre $L^{\omega}(\mathcal{B}_{\mathcal{M}}) = \{L(s_0)L(s_1)\ldots \mid s_0, s_1, \ldots$ végtelen út $\mathcal{M}$-ben, $s_0 \in S_0\}$:

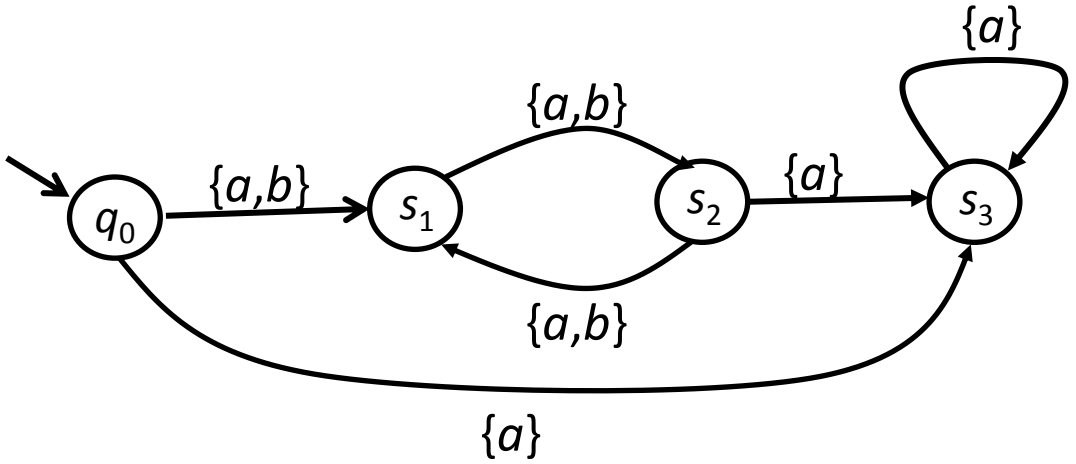
- $Q = S \cup \{q_0\}$, $q_0 \notin S$,
- $\Sigma = \mathcal{P}(AP)$,
- $\forall q, p \in S$-re $(q, a, p) \in \delta \leftrightarrow \exists t \in T$, $\alpha(t) = q$, $\beta(t) = p$, $a = L(p)$,
- $\forall q \in S_0$, $\forall a \in \Sigma$-ra $(q_0, a, q) \in \delta \leftrightarrow a = L(q)$,
- $I = \{q_0\}$,
- $F = \{Q\}$.

példa: $\mathcal{M}$ átmeneti rendszerhez a $\mathcal{B}_{\mathcal{M}}$ Büchi-automata megadása

$\mathcal{M}$:



$\mathcal{B}_{\mathcal{M}}$:



$F = \{\{q_0, s_1, s_2, s_3\}\}$

2. f LTL formulához olyan $\mathcal{B}_f$ Büchi-automata megadása, melyre

$L^{\omega}(\mathcal{B}_f) = \{v \in (\mathcal{P}(AP))^{\omega} \mid \exists c$ végtelen út, c lenyomata v, $c \models f\}$

- f formulában csak $\{\wedge, \neg, \mathbf{X}, \mathbf{U}\}$-beli operátorok
- $\models \subseteq ((\mathcal{P}(AP))^{\omega} \times$ LTL) reláció a legszűkebb olyan halmaz, melyre $\forall v = v_0 v_1 \ldots \in (\mathcal{P}(AP))^{\omega}$, $a \in AP$, g, g_1, g_2 LTL formulákra teljesülnek a következők:

$v \models \underline{1}$,

$v \models a \leftrightarrow a \in v_0$,

$v \models \neg g \leftrightarrow v \not\models g$,

$v \models g_1 \wedge g_2 \leftrightarrow v \models g_1$ és $v \models g_2$,

$v \models \mathbf{X} g \leftrightarrow v^1 \models g$,

$v \models g_1 \mathbf{U} g_2 \leftrightarrow \exists j \geq 0$, $v^j \models g_2$ és $\forall\ 0 \leq i < j$-re $v^i \models g_1$.

- *szavak*(f) = $\{v \in (\mathcal{P}(AP))^{\omega} \mid v \models f\}$
  ($v \in$ *szavak*(f) $\leftrightarrow c \models f$, ahol c végtelen út lenyomata v)
- $C(f) = \{g, \overline{g} \mid g$ az f részformulája, $\overline{g} = g'$, ha $g = \neg g'$,
  egyébként $\overline{g} = \neg g\}$
- $\mathcal{B}_f = (Q, \Sigma, \delta, I, F)$ Büchi-automata megadása, melyre
  $L^{\omega}(\mathcal{B}_f)$ = *szavak*(f):

  $Q \subseteq \mathcal{P}(C(f))$, melyre $\forall q \in Q$-ra teljesülnek:
  - $\forall g \in C(f)$-re, ha $g \notin q$, akkor $\neg g \in q$,
  - $\forall\ g_1 \wedge g_2, g \in C(f)$-re:
    $g_1 \wedge g_2 \in q \leftrightarrow g_1 \in q$ és $g_2 \in q$,
    $g \in q \leftrightarrow \neg g \notin q$,

- $\forall\ g_1 \mathbf{U} g_2 \in C(f)$-re:
  ha $g_2 \in q$, akkor $g_1 \mathbf{U} g_2 \in q$,
  ha $g_1 \mathbf{U} g_2 \in q$ és $g_2 \notin q$, akkor $g_1 \in q$,

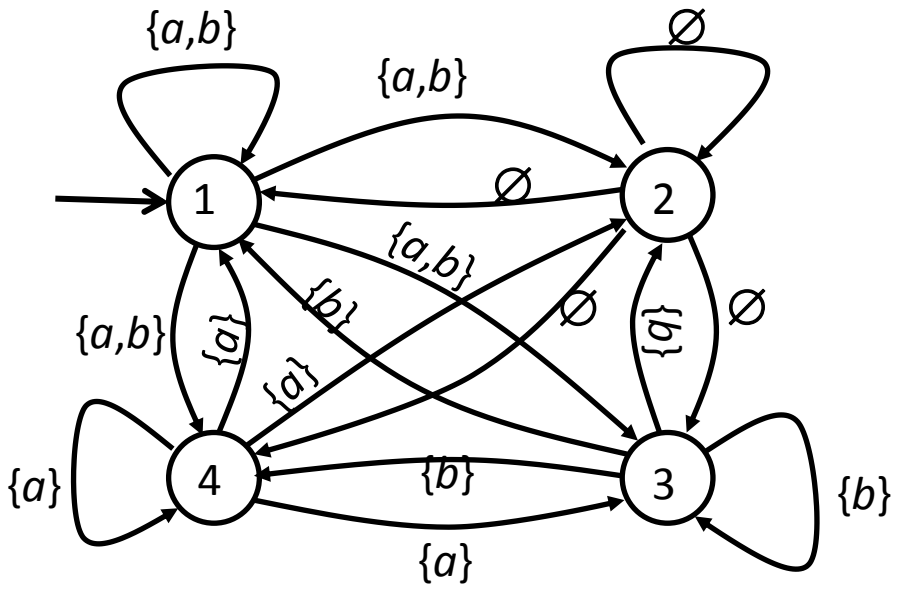
$\Sigma = \mathcal{P}(AP)$,

δ átmenetek halmaza:
$\forall\ a \in \Sigma$, q, $r \in Q$-ra $(q, a, r) \in \delta \leftrightarrow a = \{p \in AP \mid p \in q\}$ és
- ha $\mathbf{X} g \in C(f)$, akkor $\mathbf{X} g \in q \leftrightarrow g \in r$,
- ha $g \mathbf{U} h \in C(f)$, akkor $g \mathbf{U} h \in q \leftrightarrow$
  ($h \in q$ vagy $g \in q$ és $g \mathbf{U} h \in r$),

$I = \{q \in Q \mid f \in q\}$,

$F = \{F_{g \mathbf{U} h} \mid g \mathbf{U} h \in C(f)\}$, ahol
$F_{g \mathbf{U} h} = \{q \in Q \mid h \in q$ vagy $g \mathbf{U} h \notin q\}$.

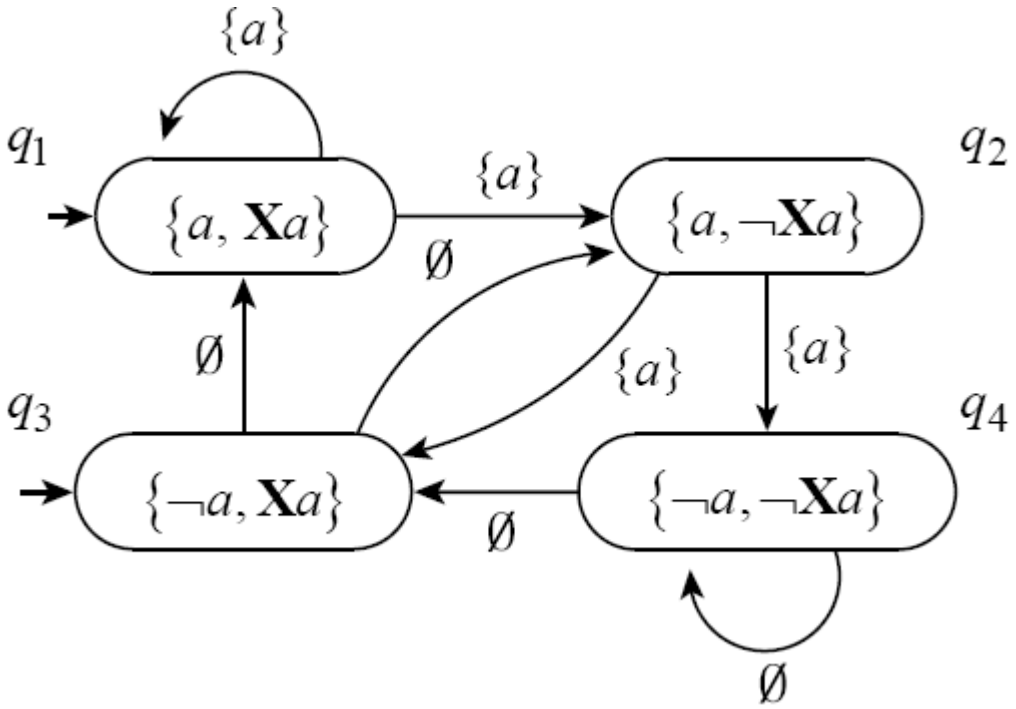
példa: $a \wedge b$ LTL formulához $\mathcal{B}_{a \wedge b}$ Büchi-automata megadása:



1 = {a,b, $a \wedge b$},
2 = {$\neg a$,$\neg b$, $\neg (a \wedge b)$},
3 = {$\neg a$,b, $\neg (a \wedge b)$},
4 = {a,$\neg b$, $\neg (a \wedge b)$}

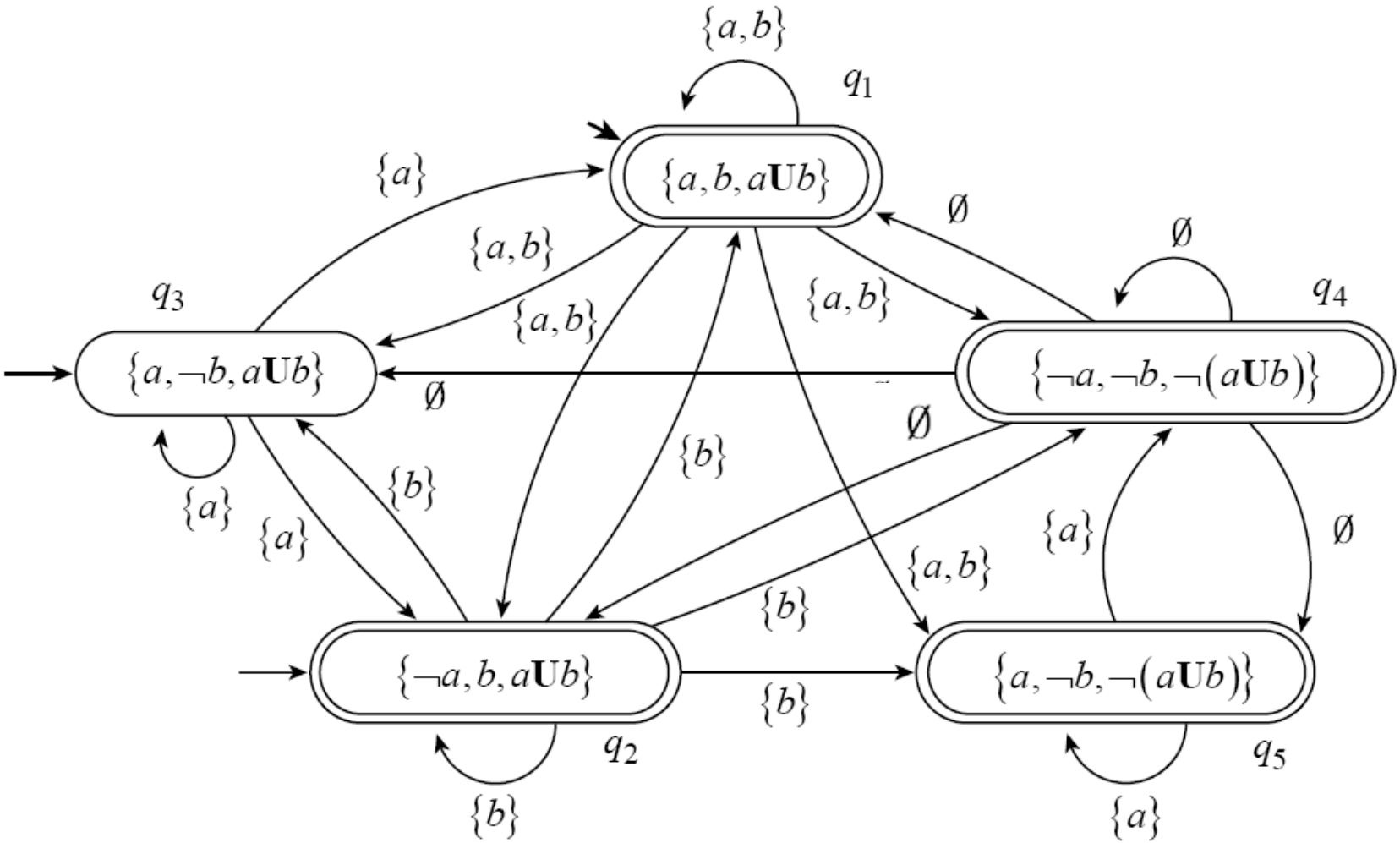
A F végállapotok halmaza legyen {{1, 2, 3, 4}} (vagy ∅) így az 1 állapotból induló minden végtelen számítási sorozathoz tartozó v végtelen szót a $\mathcal{B}_{a \wedge b}$ elfogad és $v \in szavak(a \wedge b)$.

példa: **X***a* LTL formulához $\mathcal{B}_{\mathbf{X}a}$ Büchi-automata megadása



F legyen {{q_1, q_2, q_3, q_4}} így q_1 és q_3-ból induló minden végtelen számítási sorozathoz tartozó *v* végtelen szót a $\mathcal{B}_{\mathbf{X}a}$ elfogad és $v \in szavak(\mathbf{X}a)$, mivel $v^1 \models a$.

példa: $a\mathbf{U}b$ LTL formulához $\mathcal{B}_{a\mathbf{U}b}$ Büchi-automata megadása



$F = \{F_{a\mathbf{U}b}\}$, ahol $F_{a\mathbf{U}b} = \{q_1, q_2, q_4, q_5\}$

$\mathcal{B}_f$ Büchi-automata megkonstruálásának hely és időigénye $2^{|f|}$, mivel $Q \subseteq \mathcal{P}(C(f))$.

3. $\mathcal{M} \models f$ reláció eldöntése a $\mathcal{B}_{\mathcal{M}}$ és $\mathcal{B}_f$ Büchi-automaták ismeretében visszavezethető

$L^{\omega}(\mathcal{B}_{\mathcal{M}}) \subseteq L^{\omega}(\mathcal{B}_f)$ vizsgálatra:

$$L^{\omega}(\mathcal{B}_{\mathcal{M}}) \subseteq L^{\omega}(\mathcal{B}_f) \leftrightarrow L^{\omega}(\mathcal{B}_{\mathcal{M}}) \cap \overline{L^{\omega}(\mathcal{B}_f)} = \emptyset$$
$$\leftrightarrow L^{\omega}(\mathcal{B}_{\mathcal{M}}) \cap L^{\omega}(\mathcal{B}_{\neg f}) = \emptyset.$$

LTL automata-elméleti alapú modell-ellenőrzés időbonyolultsága: $\mathcal{O}(|S| \cdot 2^{|f|})$

HML formulákra tabló-módszer alapú modell-ellenőrzés

- $\mathcal{M} = (S, T, \alpha, \beta, S_{x_1}, \ldots, S_{x_n}, T_{y_1}, \ldots, T_{y_m})$ *(X, Y)*-paraméteres átmeneti rendszer, mely $\lambda: T \rightarrow Y$ leképezés szerint *Y* felett címkézett;
- $AP = \{P_x \mid x \in X\}$;
- *f* HML formula, melyben negáció operátor ($\neg$) csak logikai konstansra vagy *AP*-beli atomi kijelentésre van;
- $R(f) = \{g \mid g$ az *f*-nek részformulája$\}$;
- $V(f) = \{s \vdash g \mid s \in S, g \in R(f)\} \cup \{0, 1\}$;
- $A(s, a) = \{q \in S \mid \exists t \in T, \alpha(t) = s, \beta(t) = q, \lambda(t) = a\}$, $\forall s \in S$, $a \in Y$ elemre;
- $c(v)$ a *v* csúcs címkéje a $\mathcal{P}(V(f))$ elemeivel címkézett szemantikus fában;

feladat: $\mathcal{M}, s_0 \models f$ reláció eldöntése $s_0 \in S$-re

f formulához a *t* szemantikus fa konstruálásának lépései:

0. *t*-nek kezdetben egyetlen csúcsa van, $c(t) = \{s_0 \vdash f\}$.

1. ha *t*-ben $\exists v$ levélcsúcs, $0 \notin c(v)$, akkor válasszuk $c(v)$ valamely $s \vdash g$ elemét

- $g = p$ vagy $g = \neg p$, $p \in AP$, akkor $s \vdash g$ helyébe 1 kerül, ha $\mathcal{M}, s \models g$, egyébként 0;
- $g = \underline{0}$ vagy $g = \underline{1}$, akkor $s \vdash g$ helyébe g kerül;
- $g = g_1 \wedge g_2$, akkor v-nek egy v_1 leszármazottja lesz és $c(v_1) = (c(v) - \{s \vdash g\}) \cup \{s \vdash g_1, s \vdash g_2\}$;

- $g = g_1 \vee g_2$, akkor v-nek két leszármazottja, v_1, v_2 lesz
  és $c(v_1) = (c(v) - \{s \vdash g\}) \cup \{s \vdash g_1\}$,
  $c(v_2) = (c(v) - \{s \vdash g\}) \cup \{s \vdash g_2\}$;
- $g = [a]r$ és $A(s, a) = \{s_1,\ldots,s_k\} \neq \emptyset$, akkor v-nek egy v_1 leszármazottja lesz és
  $c(v_1) = (c(v) - \{s \vdash g\}) \cup \{s_1 \vdash r, \ldots, s_k \vdash r\}$,
  ha $A(s, a) = \emptyset$, akkor $c(v) = (c(v) - \{s \vdash g\}) \cup \{1\}$;
- $g = \langle a \rangle r$ és $A(s, a) = \{s_1,\ldots,s_k\} \neq \emptyset$, akkor v-nek k leszármazottja lesz, $v_1,\ldots, v_k$ és
  $c(v_i) = (c(v) - \{s \vdash g\}) \cup \{s_i \vdash r\}$, $1 \leq i \leq k$,
  ha $A(s, a) = \emptyset$, akkor $c(v) = (c(v) - \{s \vdash g\}) \cup \{0\}$.

2. szemantikus fa építése befejeződik :
minden levél címkéje {1} vagy tartalmazza a 0-t, egyébként a fa építése az 1. lépéstől folytatódik.

állítások:

- a szemantikus fa építése véges lépésben befejeződik: 1. lépésben egy újabb *v'* levélcsúccsal való bővítés esetén a *c*(*v'*) a *c*(*v*)-től annyiban tér el, hogy valamely *s*⊢ *g* helyére *s*⊢ *p* kerül, ahol *p* a *g* valódi részformulája;
- a szemantikus fa konstruálás algoritmus nem-determinisztikus (egy relációhoz több szemantikus fa is adható);
- ha *v* csúcsnak *v'* leszármazott csúcsa és *c*(*v'*)-ben levő minden *s*⊢ *p* relációnak megfelelő $\mathcal{M}$, *s* ⊨ *p* reláció teljesül, vagy *c*(*v'*) = {1}, akkor *c*(*v*)-ben is minden reláció teljesül;

Zárt szemantikus fa: minden levél csúcs címkéje tartalmaz 0-t;

Nyitott szemantikus fa: ha nem zárt, vagyis legalább egy levél csúcs {1};

$\forall\ s \in S$, f HML formula esetén az alábbi 3 állítás ekvivalens:

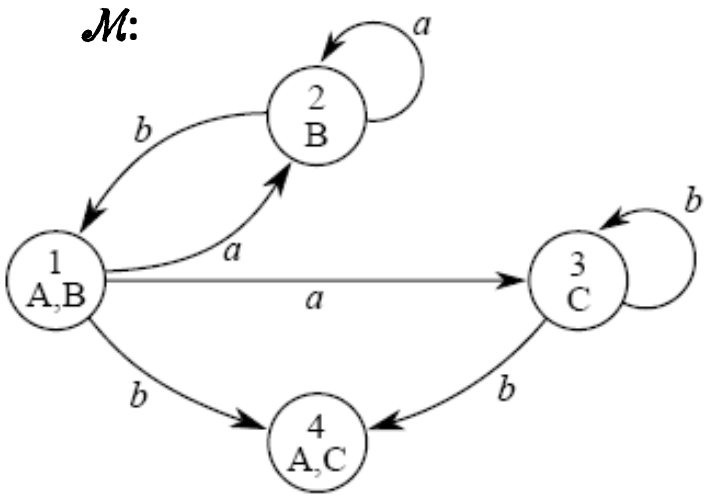
(1) $\mathcal{M}, s \models f$ teljesül.

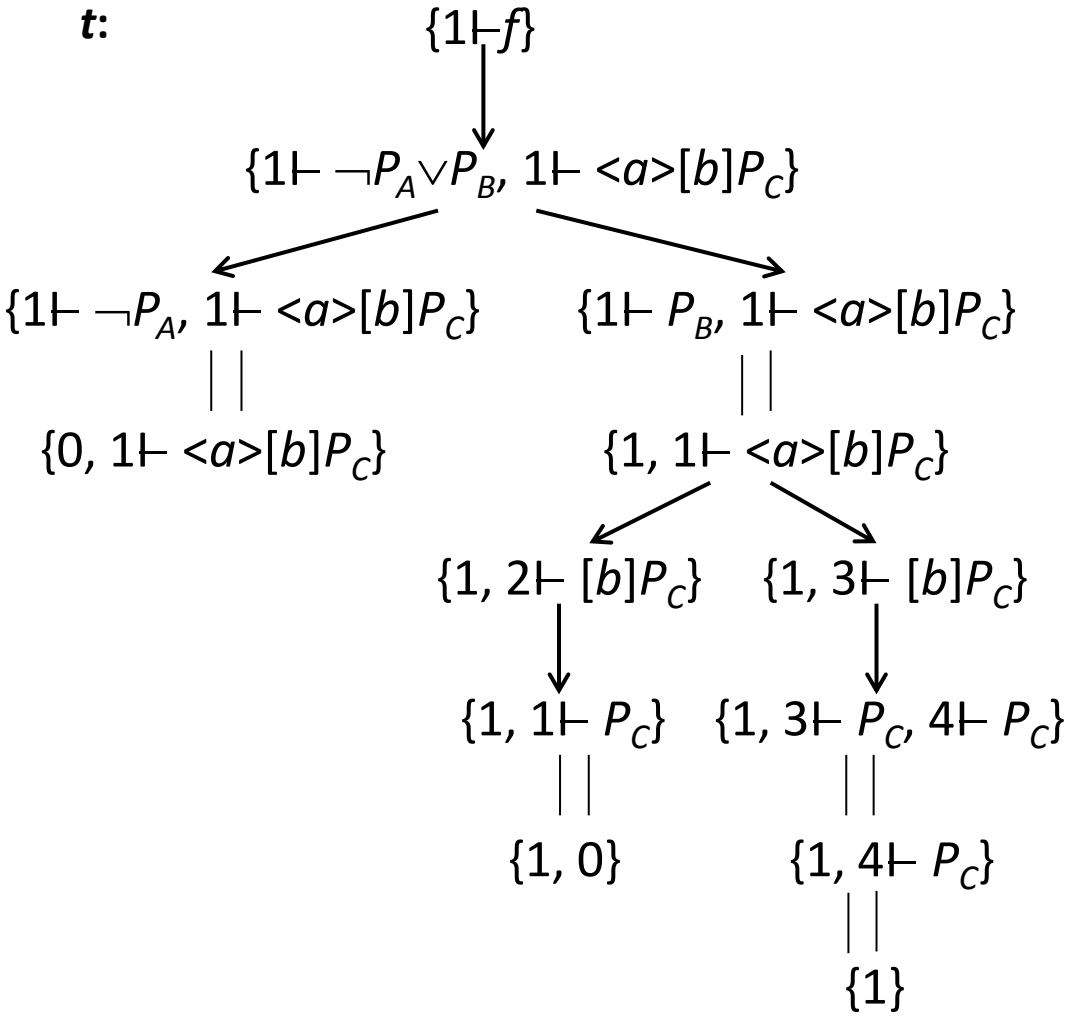
(2) $\{s \vdash f\}$ címkéjű csúcshoz konstruálható nyitott szemantikus fa.

(3) $\{s \vdash f\}$ címkéjű csúcshoz konstruálható minden szemantikus fa nyitott.

példa: $\mathcal{M}$ felett $f = (\neg P_A \vee P_B) \wedge \langle a \rangle [b] P_C$ formulához

t szemantikus fa konstruálás





t fa nyitott $\Rightarrow \mathcal{M}, 1 \models f$

TCTL modell-ellenőrzés

$\mathcal{A} = (L, A, C, L_0, E, I, AP, \rho)$ véges időzített automata, a $T(\mathcal{A})$ idő-divergens átmeneti rendszer;

φ TCTL formula AP atomi kijelentések, C órahalmaz felett;

feladat: $\mathcal{A} \models \varphi$ vagyis $T(\mathcal{A}) \models \varphi$ eldöntése

módszer:

$T(\mathcal{A}) \models \varphi$ eldöntése a $T(\mathcal{A})$ konfigurációs gráfjának végtelensége miatt gráf bejárással nem oldható meg, helyette visszavezetjük a problémát egy véges átmeneti rendszer feletti CTL modell-ellenőrzésre;

jelölések:

- $TCTL_{\geq 0}$ azon TCTL formulák halmaza, melyben operátor idő-paraméterként csak a $[0, \infty)$ fordul elő;
- $\mathcal{AB}(\mathcal{A})$ az $\mathcal{A}$ specifikációjában előforduló atomi órafeltételek halmaza;
- $\mathcal{AB}(\varphi)$ a φ-ben előforduló atomi órafeltételek halmaza;
- $\lfloor d \rfloor$ a d egészrésze, $\langle d \rangle$ a d törtrésze, $d \in \mathbb{R}_{\geq 0}$-re;
- c_x az $\mathcal{AB}(\mathcal{A})$ és $\mathcal{AB}(\varphi)$-ben $x \in C$-re vonatkozó órafeltételekben szereplő legnagyobb konstans;

1. lépés:

φ formulában az idő-paraméterek eliminálása

kiinduláskor: $C' = C$, $\varphi' = \varphi$;

- ha $J \neq [0, \infty)$, $\varphi' = \mathbf{EU}_J(f, g)$ vagy $\varphi' = \mathbf{AU}_J(f, g)$, akkor
  $C' = C' \cup \{z\}$, ahol z új óra,
  $\varphi' = \mathbf{EU}[(f \vee g),(z \in J) \wedge g]$, illetve $\varphi' = \mathbf{AU}[(f \vee g),(z \in J) \wedge g]$;
- ha $J = [0, \infty)$, akkor φ'-ben J idő-paraméter törlése;
- amíg φ'-ben van idő-paraméteres operátor ismétli az 1. lépést;

1. lépés eredményeként előálló φ' AP és C' feletti $TCTL_{\geq 0}$ formula;

φ' az $AP' = AP \cup \mathcal{AB}(\mathcal{A}) \cup \mathcal{AB}(\varphi')$ atomi kijelentések feletti CTL formula;

$TCTL_{\geq 0}$ beágyazható az *AP*' feletti CTL-be;

jelölések:

- $\mathcal{A} \oplus z = (L, A, C \cup \{z\}, L_0, E, I, AP, \rho)$ véges időzített automata, $z \notin C$;
- $z \notin C$, $d \in \mathbb{R}_{\geq 0}$, $v \in V(C)$,

$$v\{z:=d\}(x) = \begin{cases} v(x), \text{ ha } x \in C \\ d, \text{ ha } x = z \end{cases}$$

- $q = (\ell, v)$, $\ell \in L$, $v \in V(C)$, ekkor
  $q\{z:=d\} = (\ell, v\{z:=d\})$ $T(\mathcal{A} \oplus z)$-beli állapot;

állítás:

q $T(\mathscr{A})$-beli konfiguráció (v. állapot), $\mathbf{EU}_J(f, g)$, $\mathbf{AU}_J(f, g)$

TCTL formulákra:

- $T(\mathscr{A}), q \models \mathbf{EU}_J(f, g) \leftrightarrow T(\mathscr{A} \oplus z), q\{z:=0\} \models \mathbf{EU}[(f \vee g), (z \in J) \wedge g]$,
- $T(\mathscr{A}), q \models \mathbf{AU}_J(f, g) \leftrightarrow T(\mathscr{A} \oplus z), q\{z:=0\} \models \mathbf{AU}[(f \vee g), (z \in J) \wedge g]$.

példa:

$f_1 = \mathbf{EF}_{\leq 2} g$, ekkor $f_1' = \mathbf{EF}[(z \leq 2) \wedge g']$,

$f_2 = \neg\mathbf{AF}_{\leq 2} \neg\, g$, ekkor $f_2' = \neg\mathbf{AF}[(z \leq 2) \wedge \neg g']$.

2. lépés:

$T(\mathscr{A})$ Q konfiguráció-halmazán konfiguráció-ekvivalencia reláció (jelölése $\cong$) megadás

- f $\mathrm{TCTL}_{\geq 0}$ formula (1. lépés után feltehető)

$\cong_o$ óra-ekvivalencia reláció definiálása a $V(C)$ felett:

$v, v' \in V(C)$-re $v \cong_o v' \leftrightarrow \forall x, y \in C$-re teljesül:

- $v(x) > c_x$ és $v'(x) > c_x$ vagy $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$;
- ha $v(x) < c_x$, akkor $\langle v(x) \rangle = 0 \leftrightarrow \langle v'(x) \rangle = 0$;
- ha $v(x) \leq c_x$ és $v(y) \leq c_y$, akkor $\langle v(x) \rangle \leq \langle v(y) \rangle \leftrightarrow \langle v'(x) \rangle \leq \langle v'(y) \rangle$;

- $\cong_o$ reláció a $V(C)$ felett ekvivalencia reláció;
- óra-régiók a $V(C)/\cong_o$ halmaz elemei;
- $\forall r \in V(C)/\cong_o$, $v, v' \in r$, $g \in \mathcal{AB}(\mathcal{A}) \cup \mathcal{AB}(\varphi)$-re teljesül $v \models g \leftrightarrow v' \models g$;
- $V(C)/\cong_o$ óra-régiók halmaza véges:

$$|C|! \cdot \prod_{x \in C} c_x \leq |V(C)/\cong_o| \leq 2^{|C|} \cdot |C|! \cdot \prod_{x \in C} (2 \cdot c_x + 2)$$

jelölések:

$r_\infty = \{v \in V(C) \mid \forall x \in C\text{-re } v(x) > c_x\}$;

$[v] = \{v' \in V(C) \mid v \cong_\circ v'\}$;

$r[D \mapsto 0] = \{v[D \mapsto 0] \mid v \in r\}$, ahol $D \subseteq C$;

$r \in V(C)/\cong_\circ$ leszármazott régiója $r' \in V(C)/\cong_\circ$ (jelölése $\rightarrow(r)$):

- ha $r = r_\infty$ és $r = r'$ vagy
- ha $r \neq r_\infty$, $r \neq r'$ és $\forall v \in r$-re $\exists\, d \in \mathbb{R}_{>0}$, hogy
  $v+d \in r'$ és $\forall 0 \leq d' \leq d$-re $v+d' \in r \cup r'$;

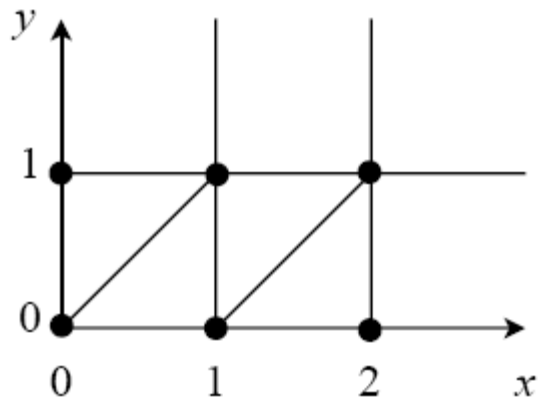
$r \in V(C)/\cong_\circ$ óra-régió kielégíti a $g \in \mathcal{AB}(\mathcal{A}) \cup \mathcal{AB}(\varphi)$ órafeltételt (jelölése $r \models g$): $\exists\, v \in V(C)$, hogy $v \in r$ és $v \models g$;

állítás:

$\forall\, r \in V(C)/\cong_\circ$, $v, v' \in r$, $D \subseteq C$-re teljesül $v[D \mapsto 0] \cong_\circ v'[D \mapsto 0]$;

példa:

$\cong_o$ reláció megadása $C = \{x,y\}$, $c_x = 2$, $c_y = 1$ esetén.



$V(C)/\cong_o$-nak 28 óra-régiója (az ábrán ezek megjelenése):
6 sarokpont, 9 nyílt szakasz, 5 félegyenes, 4 háromszög belső terület, 4 nem-korlátos terület.

példák leszármazott óra-régióra:

$\rightarrow[x = 0, y = 0] = [0 < x < 1, 0 < y < 1, x = y]$,

$\rightarrow[0 < x < 1, 0 < y < 1, x = y] = [x = 1, y = 1]$,

$\rightarrow[x = 2, y > 1] = r_\infty = [x > 2, y > 1]$,

$\rightarrow[1 < x < 2, 0 < y < 1, x\text{-}1 = y] = [x = 2, y = 1]$,

$\rightarrow[0 < x < 1, y > 1] = [x = 1, y > 1]$.

$T(\mathscr{A})$ konfiguráció halmaza (Q) felett a
$\cong$ konfiguráció-ekvivalencia reláció megadása:
$\forall\ q = (\ell, v), q' = (\ell', v') \in Q$-ra $q \cong q' \leftrightarrow \ell = \ell'$ és $v \cong_\circ v'$;

- $\cong$ reláció a $T(\mathscr{A})$ Q konfigurációs halmazán ekvivalencia reláció;
- $Q/\cong$ elemei a konfiguráció-régiók;

jelölés:
$[q]$ vagy $(\ell, [v])$: az a konfiguráció-régió, amely tartalmazza $q = (\ell, v) \in Q$ konfigurációt ;

állítás:
$\forall\ q, q' \in Q$-ra, ha $q \cong q'$, akkor $\forall a \in AP$-re $q \models a \leftrightarrow q' \models a$.

3. lépés: $RT(\mathscr{A}, \varphi) = (S, A', T, S_0, AP', \rho')$ régió-átmeneti rendszer konstruálása:

- $S = Q/\cong = \{[q] \mid q \in Q\}$;
- $A' = A \cup \{\varepsilon\}$, $\varepsilon \notin A$;
- $T \subseteq S \times A' \times S$, mely az alábbi átmeneteket tartalmazza:
  - ha $\ell \xrightarrow{g, \alpha, D} \ell' \in E$, $r \models I(\ell)$, $r \models g$ és $r[D \mapsto 0] \models I(\ell')$, ahol $r \in V(C)/\cong_o$, akkor $((\ell, r), \alpha, (\ell', r[D \mapsto 0])) \in T$,
  - ha $r \models I(\ell)$ és $\rightarrow(r) \models I(\ell)$, ahol $r \in V(C)/\cong_o$, akkor $((\ell, r), \varepsilon, (\ell, \rightarrow(r))) \in T$;
- $S_0 = \{[q] \mid q \in Q_0\}$, Q_0 a $T(\mathscr{A})$ kezdőkonfigurációinak halmaza ;

- $AP' = \mathcal{AB}(\mathcal{A}) \cup \mathcal{AB}(\varphi) \cup AP$,
- $\rho' : S \to \mathcal{P}(AP')$, $\forall s = (\ell, [v]) \in S$-re
  $\rho'((\ell, [v])) = \rho(\ell) \cup \{g \in \mathcal{AB}(\mathcal{A}) \cup \mathcal{AB}(\varphi) \mid [v] \models g\}$;

$RT(\mathcal{A}, \varphi)$ régió-átmeneti rendszerben S, A', T, AP' véges halmazok.

példa:

A következő dián látható $\mathcal{A}$ időzített átmeneti rendszer, ahol $AP = \{ki, be\}$, és a $\rho(1) = \{ki\}$, $\rho(2) = \{be\}$;

$\varphi = \mathbf{EF}_{\leq 1} ki$ TCTL formula;

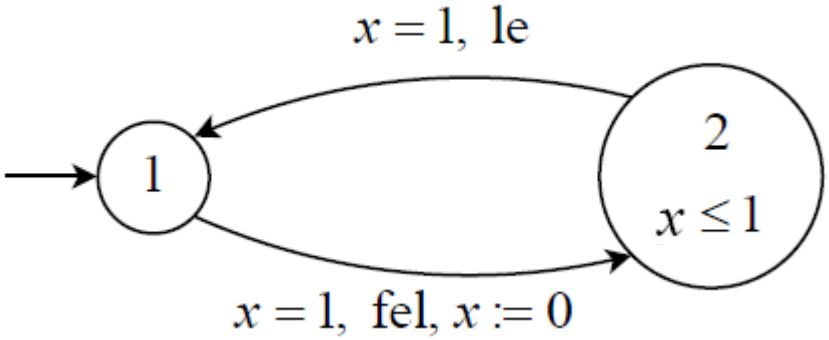
$\varphi' = \mathbf{EF}[(z \leq 1) \wedge ki]$, z új óra;

$AP' = \{ki, be, z \leq 1, x \leq 1, x = 1\}$

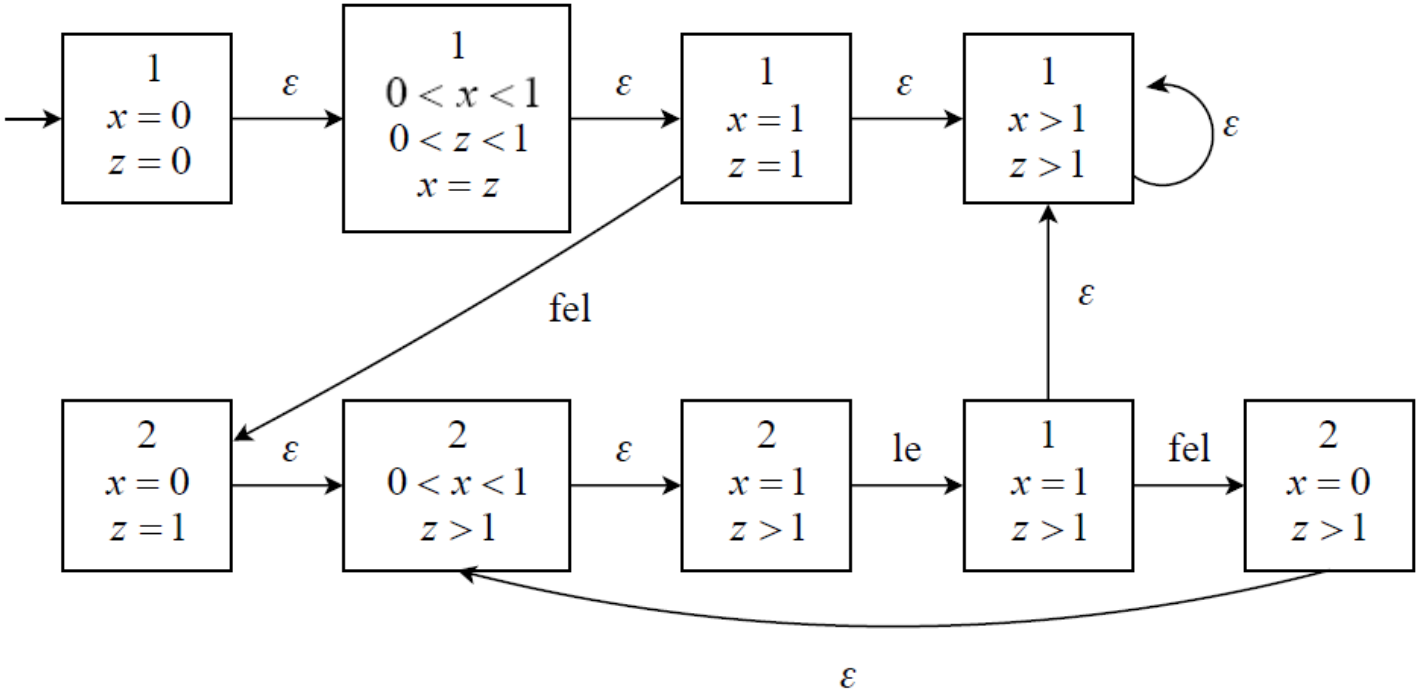
$RT(\mathcal{A} \oplus z, \varphi')$ régió-átmeneti rendszert szemlélteti a következő dia;

példa:

$\mathcal{A}$:



$RT(\mathcal{A} \oplus z, \varphi')$:



4. lépés: TCTL modell-ellenőrzés

állítás:

$\mathcal{A} \models \varphi \leftrightarrow RT(\mathcal{A}, \varphi) \models \varphi'$ a CTL szemantika szerint

módszer:

- az operátorok idő-paramétereinek eliminálása egyetlen új *z* óra bevezetésével;
- $RT(\mathcal{A} \oplus z, \varphi)$ régió-átmeneti rendszer konstruálása az $\mathcal{A} \oplus z$ időzített automata és $\mathcal{AB}(\varphi)$ (ez tartalmazza a φ-ben részformulaként előforduló atomi órafeltételeket és az összes operátor idő-paraméterként benne előforduló ($z \in J$) atomi órafeltételeket) alapján;

- jelölések az alábbi algoritmusban:

$S_R(\psi) = \{s \in S \mid R, s \models \psi\}$, ahol S az R régió-átmeneti rendszer állapotainak halmaza;

$Sub(\varphi)$: a φ részformuláinak halmaza;

$Lab(s)$: $\forall s \in S$-re tartalmazza a $\rho'(s)$ elemeit és azokat a a_ψ CTL formulákat, melyek a ψ-ből az operátorok idő-paramétereinek eliminálásával állnak elő és melyekre teljesül a $R, s \models \psi$ reláció;

$S_{CTL}(\psi)$ a CTL modell-ellenőrzési módszer alapján ψ-t kielégítő S-beli állapotok halmaza;

algoritmus:

bemenet: $\mathcal{A}$ idő-divergens időzített automata, φ TCTL formula;

kimenet: „igen”, ha $\mathscr{A} \models \varphi$, egyébként „nem”.

R:= $RT(\mathscr{A} \oplus z, \varphi)$;

forall $i \leq |\varphi|$ do

  forall $\psi \in Sub(\varphi)$ és $|\psi| = i$ do

    switch (ψ):

      $\underline{1}$: $S_R(\psi) := S$;

      a: $S_R(\psi) := \{s \in S \mid a \in Lab(s)\}$;

      $\psi_1 \wedge \psi_2$: $S_R(\psi) := \{s \in S \mid \{a_{\psi_1}, a_{\psi_2}\} \in Lab(s)\}$;

      $\mathbf{EU}_J(\psi_1, \psi_2)$: $S_R(\psi) := S_{CTL}(\mathbf{EU}((a_{\psi_1} \vee a_{\psi_2}), (z \in J) \wedge a_{\psi_2}))$;

      $\mathbf{AU}_J(\psi_1, \psi_2)$: $S_R(\psi) := S_{CTL}(\mathbf{AU}((a_{\psi_1} \vee a_{\psi_2}), (z \in J) \wedge a_{\psi_2}))$;

    endswitch

    forall $s \in S$ és $s\{z:=0\} \in S_R(\psi)$ do $Lab(s) := Lab(s) \cup \{a_\psi\}$ od

  od

od

if $S_0 \subseteq S_R(\varphi)$ then return „igen” else return „nem” fi

- $\mathcal{A} \models \varphi$ reláció eldönthető $\mathcal{O}(|\varphi| \cdot (|S|+|T|))$, mivel a CTL modell-ellenőrzést alkalmazza, ahol *S* és *T* az $RT(\mathcal{A}, \varphi)$ állapot-, illetve átmenethalmaza;
- $RT(\mathcal{A}, \varphi)$ mérete az órák számában és a φ-ben levő órafeltételekben megjelenő maximális konstansokban exponenciális;
- időigény további javítási lehetősége: a régió-átmeneti rendszernél kisebb méretű reprezentációval (konfiguráció-régiók zónákká összevonása, anélkül, hogy a modell-ellenőrzés helyessége sérülne) ;