

# Finomítási relációk és kapcsolatuk a teszteléssel

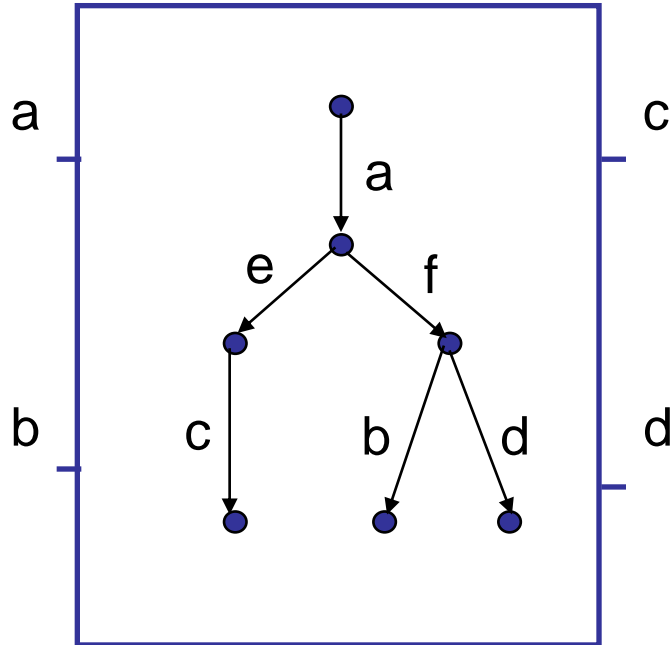
Majzik István

BME Méréstechnika és Információs Rendszerek Tanszék

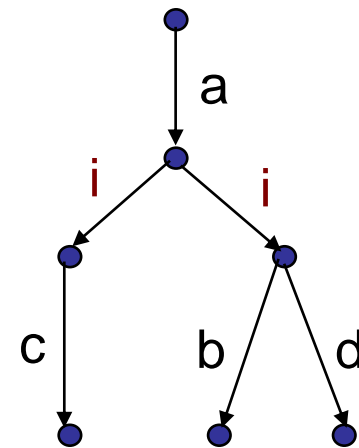
## Motiváció (ismétlés)

- Relációk modellek (LTS-ek) között:
    - Megfelelőség (ekvivalencia)
    - Finomítás (rendezés)
  - Példák finomításra modellek között:
    - Viselkedés bővül
    - Nemdeterminizmus csökken
    - Lehetséges holtpontok száma csökken
  - **Finomítási (rendezési) reláció (preorder)  $\leq$** 
    - Reflexív, tranzitív, antiszimmetrikus
- Prekongruencia reláció (precongruence)  $\leq$**
- Ha  $T1 \leq T2$ , akkor minden  $C[ ]$  környezetre  $C[T1] \leq C[T2]$
  - A beágyazás megőrzi a relációt

# Belső akciók és megfigyelhetőség (ismétlés)



Komponens  
belső viselkedés



Megfigyelhető  
viselkedés  
 $i, \tau$ : belső akció

# „Tesztelés” és „holtpont” értelmezése (ismétlés)

- **Tesztelés értelmezése LTS-eken:**
  - Rendszer mint fekete doboz, interfésszel (portok)
  - **Interakció** a környezettel: Akciók értelmezése mint szinkron kommunikáció egy-egy porton keresztül, pl.
    - Üzenet küldése és fogadása
    - Esemény és annak feldolgozása
    - (Távoli) eljárás hívása
- **Deadlock (holtpont) értelmezése a tesztelésben:**
  - A környezet egy interakciót indít, de arra a rendszer **nem reagál** (interakció nem jön létre)
    - Üzenet küldése vagy fogadása nem történik meg
    - Eseményre nincs reakció
    - Eljáráshívás nem valósulhat meg
  - Teszt **„hibázik”**: A kívánt interakció nem lehetséges

# May preorder - lehetséges viselkedés: Definíció

- **Jelölések:**

$\beta \in (Act - \tau)^*$  megfigyelhető akciószekvencia ( $\tau$  törlése)

$s \xRightarrow{\beta} s'$  ha  $\exists \alpha \in Act^*: s \xrightarrow{\alpha} s'$  és  $\beta = \hat{\alpha}$

$\Delta(s)$  az  $s$ -ből induló megfigyelhető akciósorozatok halmaza:

$$\Delta(s) = \left\{ \beta \mid \exists s' : s \xRightarrow{\beta} s' \right\}$$

- **Rendezési reláció:**

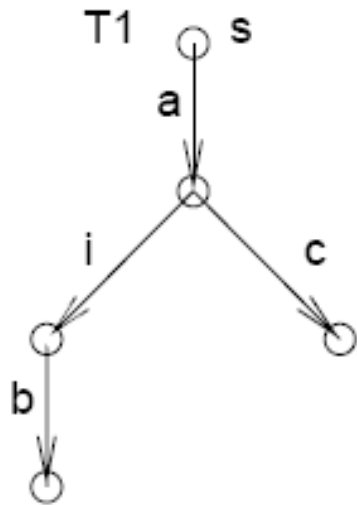
$T_1$  és  $T_2$  LTS-ekre, melyek kezdőállapota  $s_1$  és  $s_2$ ,  
és azonos  $Act$  akcióhalmazra:

$$T_1 \leq_{\Delta} T_2 \text{ a.cs.a. } \Delta(s_1) \subseteq \Delta(s_2)$$

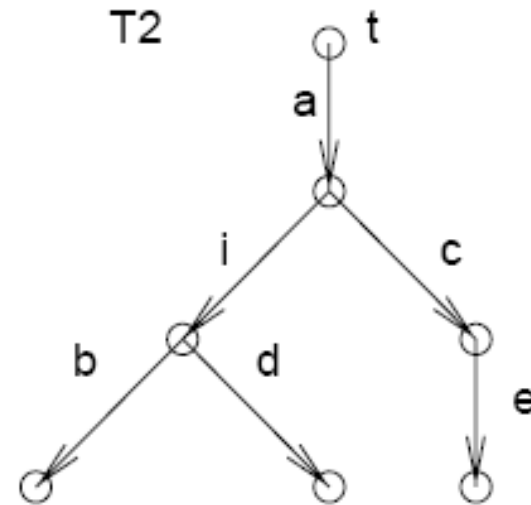
Itt  $T_2$ -ben több a megfigyelhető akciósorozat  
(bővül a lehetséges viselkedések – mint akciósorozatok – halmaza).

# May preorder - lehetséges viselkedés: Példa

- LTS-ek lehetséges megfigyelhető akciósorozatai:



$$\Delta(s) = \{a, ab, ac\}$$



$$\Delta(t) = \{a, ab, ac, ad, ace\}$$

# May preorder - lehetséges viselkedés: Tesztelés

- Kapcsolat a teszteléssel:  $T_1 \leq_{\Delta} T_2$  esetén:
  - Minden teszt, ami  $T_1$  esetén sikeres lehet,  $T_2$  esetén is sikeres lehet
    - De nem feltétlenül: nondeterminisztikus viselkedés, illetve belső akciók miatt előfordulhat, hogy sikertelen
  - Azaz a rendezés megőrzi a lehetséges sikeres tesztek:  $T_1$  lehetséges sikeres tesztjei  $T_2$  lehetséges sikeres tesztjei között vannak
- A may preorder által definiált finomítás:
  - Nem „veszik el” lehetséges megfigyelhető viselkedés
- Ennek „párjaként” definiálható másik finomítás:
  - Nem „veszik el” a szükséges megfigyelhető viselkedés
  - Megkötést ad a mindig sikeres tesztekre

# Must preorder - szükséges viselkedés: Bevezetés

- Jelölések:

$s$  elutasítja  $E \subseteq Act - \{\tau\}$  akciókat, ha  $\forall e \in E : \text{nincs } s \xRightarrow{e} s'$

$s$  divergens ( $s \uparrow$ ),

ha  $\exists s_0 s_1 \dots$  végtelen szekvencia, ahol  $s = s_0$  és  $s_i \xrightarrow{\tau} s_{i+1}$

$s$  divergál  $\beta$  akcióorozaton ( $s \uparrow \beta$ ),

ha  $\exists \beta'$  prefixe  $\beta$ -nak, hogy  $s \xRightarrow{\beta'} s'$  és  $s' \uparrow$

$\langle \beta, E \rangle$  egy hibázása  $s$ -nek, ha

vagy  $s \uparrow \beta$

vagy  $\exists s' : s \xRightarrow{\beta} s'$  és  $s'$  elutasítja  $E$  elemeit

(azaz  $\beta$  közben divergál, vagy  $\beta$  után  $E$ -t elutasítja).

$F(s)$  az  $s$ -hez tartozó összes hibázás halmaza.



# Must preorder - szükséges viselkedés: Definíció

## Rendezési reláció: Hibázások rendezése

$$T_1 \leq_F T_2 \text{ a.cs.a. } F(s_1) \supseteq F(s_2)$$

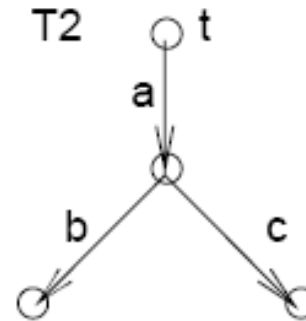
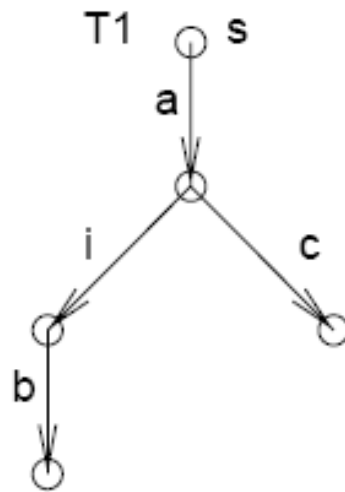
azaz  $T_2$ -ben kevesebb a hibázás, mint  $T_1$ -ben.

## Hibázások szerepe:

- Hibázás: Akció(k) elutasítása közvetlenül vagy divergencia miatt
- Kevesebb hibázás: Kevesebb elutasítási lehetőség, több sikeres akció
- A viselkedés bővítésével csökken a hibázások száma (amit eddig elutasított, ezután nem fogja)
- Nemdeterminisztikus viselkedés csökkentésével is csökkenhet a hibázások száma

# Must preorder - szükséges viselkedés: Példa

- Példa:



T1 mindig sikeres tesztjei:  $\{a,ab\}$

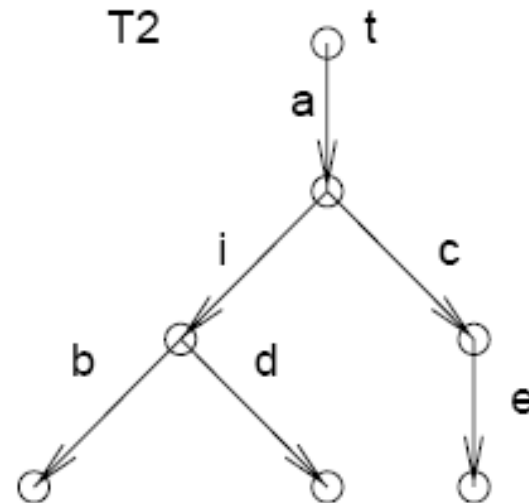
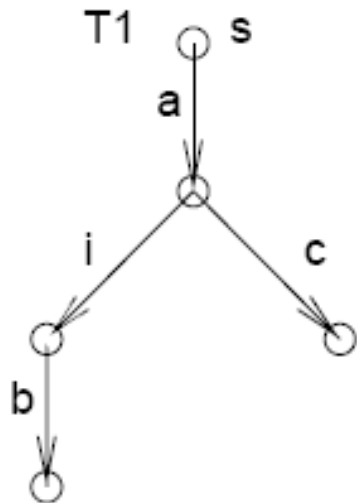
itt  $\langle a,\{c\} \rangle$  egy hibázás

T2 mindig sikeres tesztjei:  $\{a,ab,ac\}$  bővebb halmaz

itt  $\langle a,\{c\} \rangle$  nem hibázás

# Must preorder - szükséges viselkedés: Példa

- Példa:



T1 mindig sikeres tesztjei:  $\{a, ab\}$

T2 mindig sikeres tesztjei:  $\{a, ab, ad\}$  bővebb halmaz

hibázások halmaza csökken

# Must preorder - szükséges viselkedés: Tesztelés

- Kapcsolat a teszteléssel:  $T_1 \leq_F T_2$  esetén:
  - Minden teszt, ami  $T_1$  esetén sikeres,  $T_2$  esetén is sikeres
    - $T_2$ -nek kevesebb a hibázása, nem utasíthat el többet
  - A rendezés megőrzi a mindig sikeres tesztekét:  
 $T_1$  sikeres tesztjei  $T_2$  sikeres tesztjei között vannak
- A must preorder jellegzetessége:
  - A finomított LTS-ben nem „veszhet el” olyan megfigyelhető viselkedés,  
ami az absztrakt LTS-ben szerepelt
- Kapcsolat a deadlock lehetőséggel:
  - Deadlockra érzékeny rendezés

Konformancia relációk teszteléshez

# Motiváció

- Konformancia reláció elvárásai: Legyen „trace” alapú
  - Cél a **teszt során tapasztalt viselkedés** és a **specifikáció szerinti viselkedés** összevetése (hibás viselkedés azonosítása)
  - **Fekete doboz teszteléshez: Bemenetek és kimenetek**, valamint **belső (láthatatlan) akciók** megkülönböztetése
  - **Bemenetek és kimenetek átlapolódása** ne legyen megkötött
  - **Kimeneti akció hiánya** jelentsen különbségtételt (ha nem engedi meg a specifikáció a hiányt, akkor hibát jelent)
  - **Nemdeterminisztikus viselkedés** legyen kezelhető
- Az LTS-nél finomabb matematikai modell kell
  - **Akciótípusok megkülönböztetése**
    - **Bemeneti akciók:** Teszt vezérli
    - **Kimeneti akciók:** Megfigyelhető kívülről
    - **Belső (láthatatlan) akció:** A környezet (teszt) nincs rá hatással
  - **Tétlen (quiescent) állapot:**
    - Nincs kimeneti vagy belső akcióval címkézett átmenet belőle  
→ Csak bemeneti akcióval címkézett átmenet indul belőle

# Az IOLTS formalizmus

- Input-Output Labelled Transition System (IOLTS):

$$IOLTS = (S, Act, \rightarrow, s_0)$$

$S$  az állapotok halmaza,  $s_0$  kezdőállapot

$Act$  az akciók halmaza:  $Act = Act_{in} \cup Act_{out} \cup \{\tau\}$

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

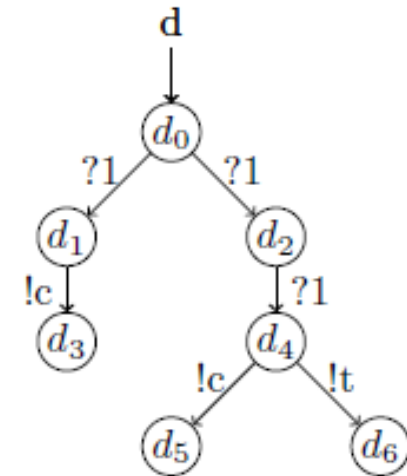
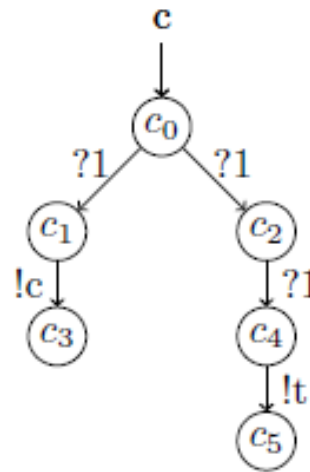
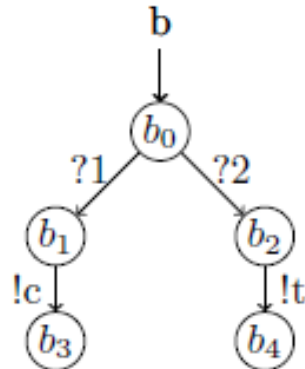
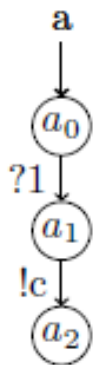
$Act_{in}$  bemeneti,  $Act_{out}$  kimeneti akciók,  $\tau$  belső akció

- Az IOLTS tulajdonságainak definíciói:
  - Teljes, ha minden állapotban minden akcióval van átmenet
  - Bemenetre teljes (weakly input enabled), ha minden állapotban minden bemenettel van átmenet, esetleg  $\tau^*$  után
    - Implementáció jellemzője: minden bemenetet kezel valahogyan
  - Determinisztikus, ha minden megfigyelhető akciósorozat esetén csakis egy elérhető állapot van

# IOLTS példák

## Italautomata IOLTS-ek:

- $Act_{in}=\{1,2\}$  bemenetek (pénzbedobás)
  - Jelölés: ? Prefix: ?1, ?2
- $Act_{out}=\{c,t\}$  kimenetek (kávé vagy tea)
  - Jelölés: ! prefix: !c, !t





# További jelölések és átalakítások

- Jelölések:

- $\beta$  megfigyelhető akciósorozata a  $T$  IOLTS modellnek
- $\Delta(T)$  egy  $T$  modell megfigyelhető akciósorozatai
- $In(s)$  az  $s$  állapotból induló bemeneti akciók halmaza
- $Out(s)$  az  $s$  állapotból induló megfigyelhető kimeneti akciók halmaza ( $S$  állapothalmazra is kiterjeszthető:  $Out(S)$ )
- Elérhető állapotok: „after” operátorral

$$s \text{ after } \beta = \left\{ s' \mid s \xRightarrow{\beta} s' \right\} \quad S \text{ after } \beta = \bigcup_{s \in S} (s \text{ after } \beta)$$

- A „tétlenség” uniformizált kezelése:

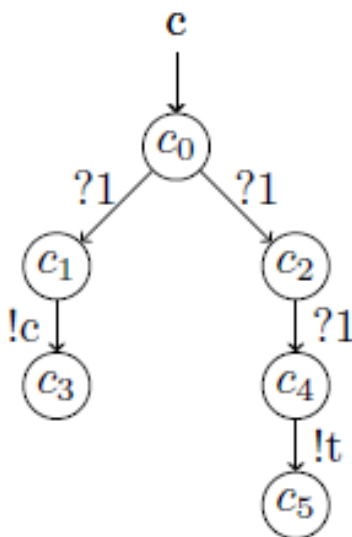
- A tétlen állapotok (csak bemenetre vár) kapnak egy hurokátmenetet a speciális  $\delta$  akcióval címkézve
- Így előáll egy módosított automata  $T_\delta$
- Ebben a tétlenség  $\delta$  kimeneti akcióként megfigyelhető

# Példa: IOLTS kiegészítése a tétlenség jelzésével

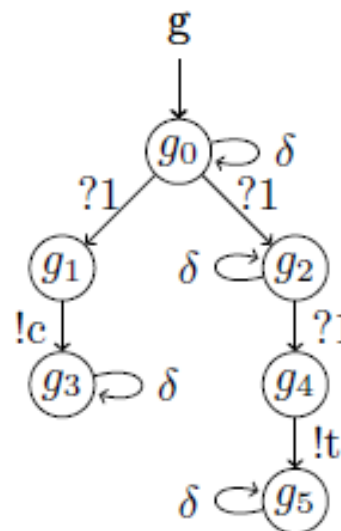
## Ital automata IOLTS:

- $Act_{in}=\{1,2\}$  bemenetek (pénzbedobás), ? prefix
- $Act_{out}=\{c,t\}$  kimenetek (kávé vagy tea), ! prefix

Ha egy állapotban nincs kimeneti akció, akkor egy  $\delta$  hurokátmenet kerül rá:



kiegészítése a tétlenséggel:

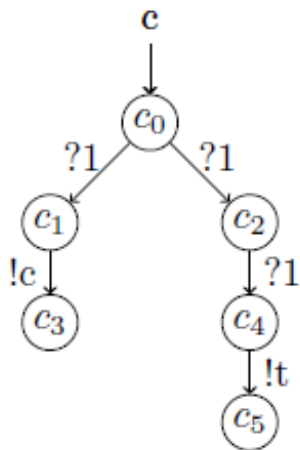


# Példa: IOLTS bemenetre teljessé tétele

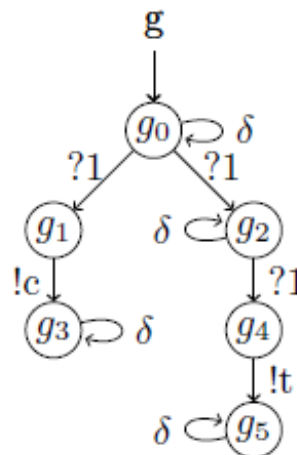
Ital automata IOLTS:

- $Act_{in}=\{1,2\}$  bemenetek (pénzbedobás), ? prefix
- $Act_{out}=\{c,t\}$  kimenetek (kávé/tea), ! prefix

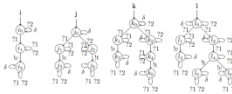
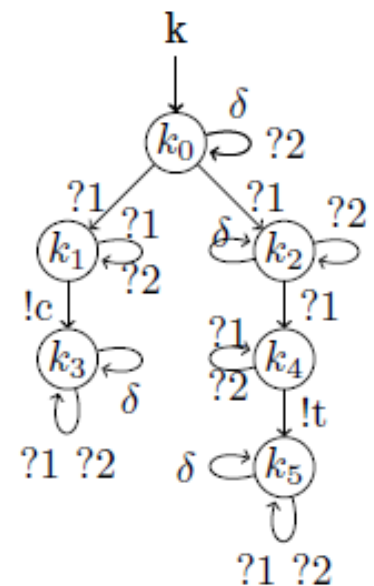
Hurokátmenet van a kimaradó bemenetekre:



kiegészítése  
a tétlenséggel:



majd  
bemenetre  
teljessé  
tételé:



# Egy IOLTS ekvivalencia reláció: A k-ekvivalencia

- Elemek:
  - $T_\delta$  IOLTS mint „specifikáció”
  - $M_\delta$  IOLTS mint „implementáció”
  - Bemenetekre következnek a kimenetek (reakció)
- Definíció:
  - A „specifikációban” és az „implementációban” azonos bemeneti sorozat mellett azonos kimenetek sorozata következik be az első  $k$  lépésre
- Jellegzetességek
  - Egyszerű reláció
  - Szigorú a teszteléshez (a  $k$  lépésszám keretén belül)
    - Szűkítés, bővítés nem megengedett

# Egy IOLTS konformancia reláció: Az IOCO

- Elemek:
  - $T_\delta$  IOLTS mint „specifikáció” (elvárt viselkedés)
  - $M_\delta$  IOLTS mint „implementáció”, ami bemenetre teljessé tett
  - Lehetséges akciók halmazai azonosak
- Definíció:  $M \text{ ioco } T$  („M ioco konform a T specifikációval”)  
 $M_\delta$  és  $T_\delta$  esetén minden, a specifikációban felvehető megfigyelhető akciószekvenciára igaz: Az így elérhető állapotokban az implementáció által nyújtott kimeneti akciók részalmazát képezik a specifikáció által nyújtott kimeneti akcióknak

$$\forall \beta \in \Delta(T_\delta) : \text{Out}(s_{0,M_\delta} \text{ after } \beta) \subseteq \text{Out}(s_{0,T_\delta} \text{ after } \beta)$$

Megfigyelhető  
 $\beta$  akció-  
szekvencia a  
specifikációban

Az implementáció  
megfigyelhető  
kimeneti akciói  
 $\beta$  után

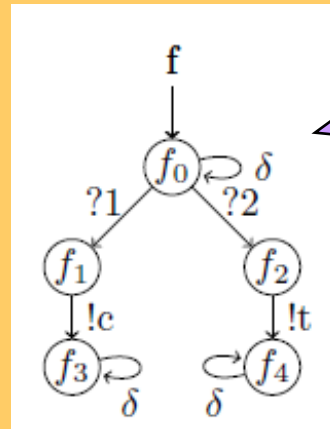
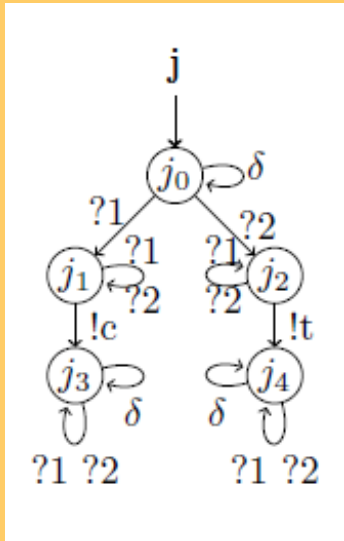
A specifikáció  
megfigyelhető  
kimeneti akciói  
 $\beta$  után

# Az IOCO jellegzetességei

- **Definíció:**
  - Formálisan: Minden, a **specifikációban felvehető megfigyelhető akciószekvenciára igaz: az így elérhető állapotokban az implementáció által nyújtott kimeneti akciók** részhalmazát képezik a **specifikáció által nyújtott kimeneti akcióknak**
  - Informálisan: Az implementáció „**befér**” a specifikáció keretei közé
    - A specifikáció a megengedhető viselkedés **kereteit** rögzíti
- **Mit enged meg?**
  - Szűkített viselkedés: Az implementációban a specifikációhoz képest kevesebb kimeneti akció jelenhet meg
    - Pl. részleges implementáció, vagy nondeterminizmus egyféle feloldása
  - Bővített viselkedés: A specifikációban **nem szereplő** akciószekvenciára megjelenhet az implementációban plusz kimenet
    - Pl. nem teljes a specifikáció (bizonyos akciószekvenciákra)
- **Mit nem enged meg?**
  - A specifikációban rögzített akciószekvenciára nem bővíthető a viselkedés (a kimenetek halmaza), tehát nem történhet „más”

# Példák IOCO konformanciára

iooco

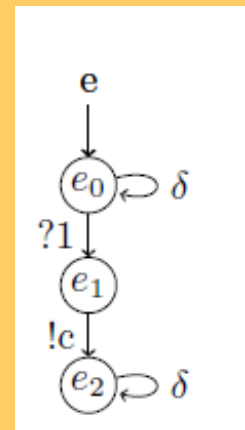
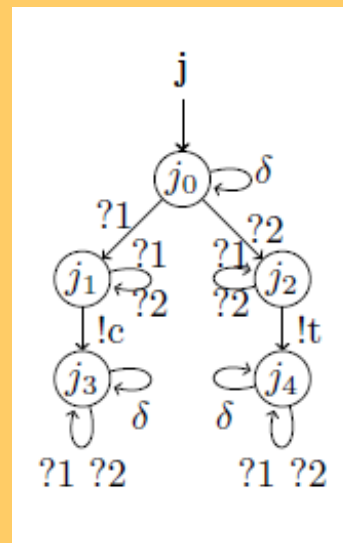


specifikáció

A megfigyelhető kimeneti akciókat kell nézni minden megfigyelhető akció-szekvencia után

Az implementáció új szekvenciákat is megenged, de a specifikációban lévő szekvenciákra itt megőrzi a viselkedést

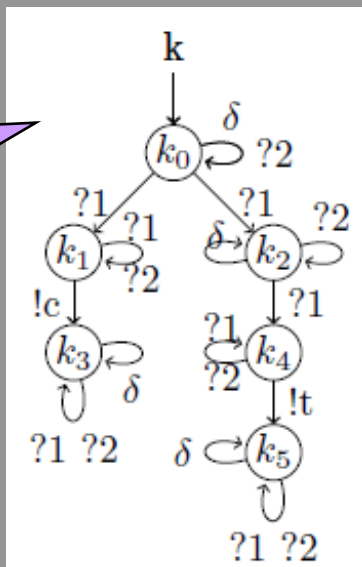
iooco



specifikáció

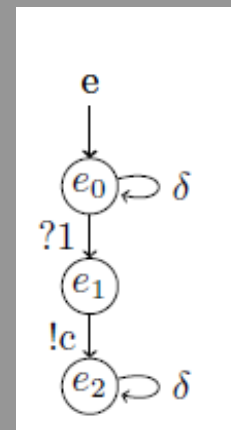
# Példák nem IOCO konformanciára

Az implementáció azokra a szekvenciákra bővíti a viselkedést, amik a specifikációban vannak



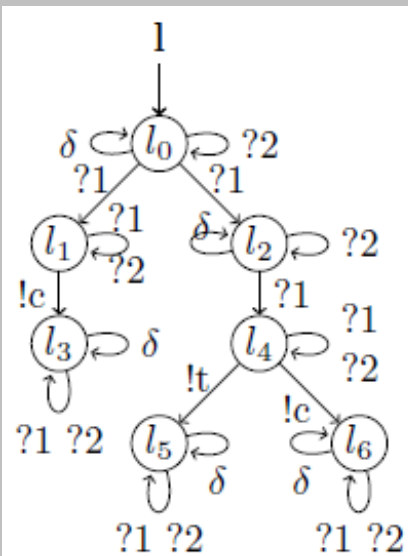
$k_0$  after  $?1 = \{!c, \delta\}$

not ioco

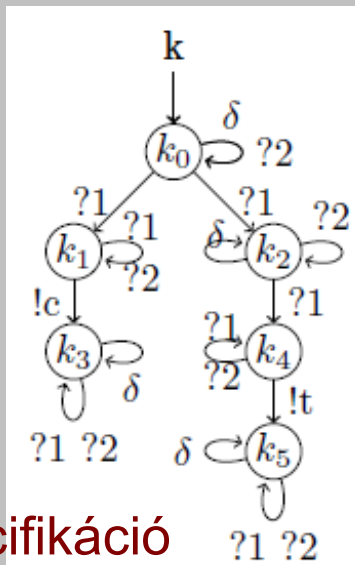


specifikáció

$e_0$  after  $?1 = \{!c\}$



not ioco



specifikáció

$k_0$  after  $\langle ?1, \delta, ?1 \rangle$

$l_0$  after  $\langle ?1, \delta, ?1 \rangle$



# Összefoglalás

- Ekvivalenciák: Verifikációhoz

- Trace:  $T \approx_{\Lambda} T'$  a.c.s.a.  $\Lambda(s) = \Lambda(s')$
- Erős biszimuláció:  $T \sim T'$  a.c.s.a.  $\exists B : (s, s') \in B$
- Megfigyelési ekvivalencia:  $T \approx T'$  a.c.s.a.  $\exists WB : (s, s') \in WB$

- Rendezések: Modellfinomításhoz

- May (lehetséges viselkedés):  $T \leq_{\Delta} T'$  a.c.s.a.  $\Delta(s) \subseteq \Delta(s')$
- Must (szükséges viselkedés):  $T \leq_F T'$  a.c.s.a.  $F(s) \supseteq F(s')$

- Konformancia reláció: Teszteléshez

- k-ekvivalencia
- Input-output konformancia (IOCO)