

Modellellenőrzés

dr. Majzik István

BME Méréstechnika és Információs Rendszerek Tanszék

Motiváció: Mit szeretnénk elérni?

- Alacsony szintű formalizmusok (KS, LTS, KTS, TA)
- Magasabb szintű formalizmusok

Temporális logikák:
PLTL, CTL, CTL*

Rendszer modellje

Követelmény megadása

Automatikus
modell ellenőrző

OK

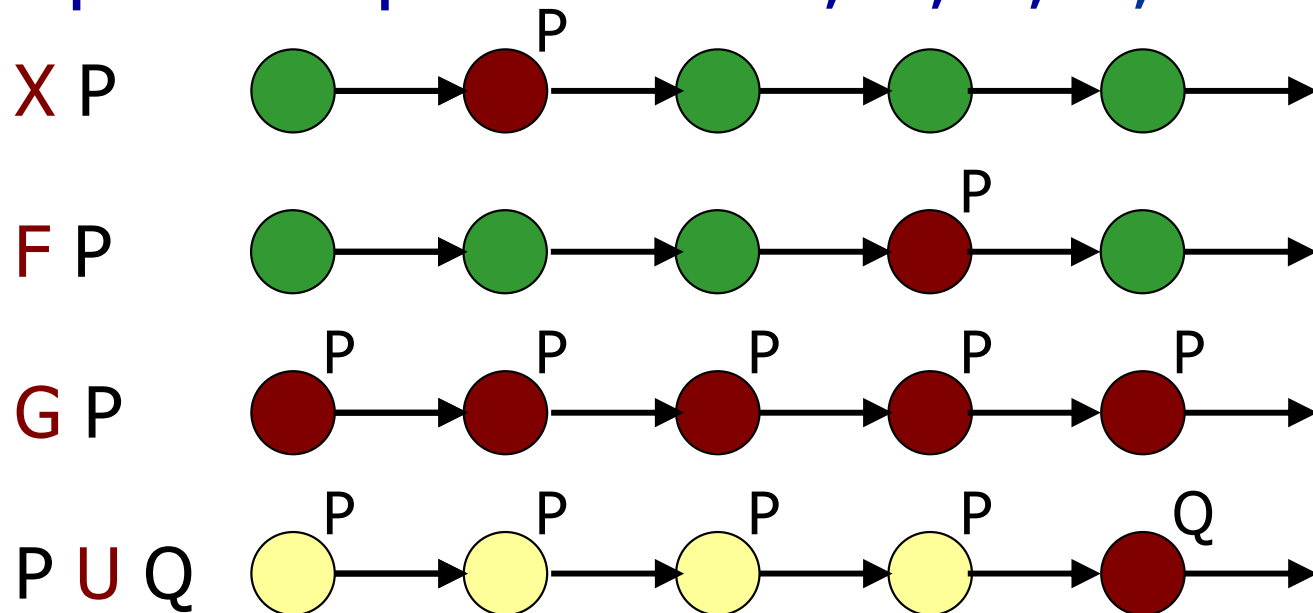
Ellenpélda

Ismétlés: Lineáris idejű temporális logika: PLTL

PLTL (Propositional Linear Time Temporal Logic)

p, q, r, \dots kifejezések konstruálása:

- Atomi kijelentések (AP elemei): P, Q, \dots
- Boole logikai operátorok: $\wedge, \vee, \neg, \Rightarrow$
 \wedge : És, \vee : Vagy, \neg : Negálás, \Rightarrow : Implikáció
- Temporális operátorok: F, G, X, U , informálisan:



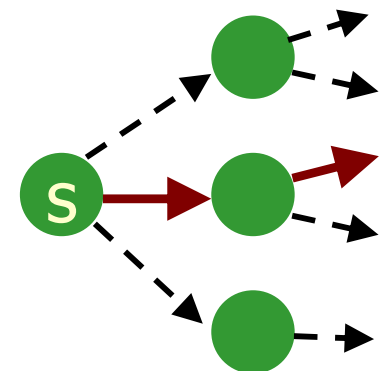
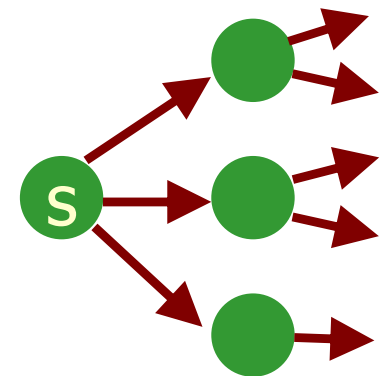
Ismétlés: Elágazó idejű temporális logika: CTL*

- Útvonalakon értelmezett operátorok (mint PLTL):

- $F p$, $G p$, $X p$, $p U q$

- Állapotokon értelmezett operátorok:

- **A**: „for All futures”, minden lehetséges útra az adott állapotból kiindulva
- **E**: „Exists future”, „for some future”, legalább egy útra az adott állapotból kiindulva



Ismétlés: Elágazó idejű temporális logika: CTL

- **Állapotokon értelmezhető összetett operátorok:**
 - **EX** p : létezik útvonal, aminek következő állapotán p
 - **EF** p : létezik útvonal, aminek egy állapotán p
 - **EG** p : létezik útvonal, aminek minden állapotán p
 - **E**($p \cup q$): létezik útvonal, amin p amíg q

 - **AX** p : minden útvonal következő állapotán p
 - **AF** p : minden útvonal egy-egy elérhető állapotán p
 - **AG** p : minden útvonal minden állapotán p
 - **A**($p \cup q$): minden útvonalon p amíg q

Az előadás áttekintése

Hogy működik a modellellenőrzés?

- A modellellenőrzés technikái
 - PLTL modellellenőrzés: **Tabló módszer**
 - CTL modellellenőrzés: **Szemantika alapú módszer**

Miért jó ezt tudni?

- Lehetőségek felmérése
 - Korlátok (ellenőrizhető modellek)
 - Optimalizálási lehetőségek (ld. következő előadás!)
- Érdekes alkalmazások
 - Automatikus teszt generálás
 - Alkalmazás a hardver tervezésben

PLTL modellellenőrzés: Tabló módszer

A modellellenőrzés feladata

Ha nincs útvonal megadva, akkor a kezdőállapotból induló minden útra ellenőriz

Kripke struktúra M

PLTL kifejezés p

Modellellenőrző
 $M, \pi \models p ?$

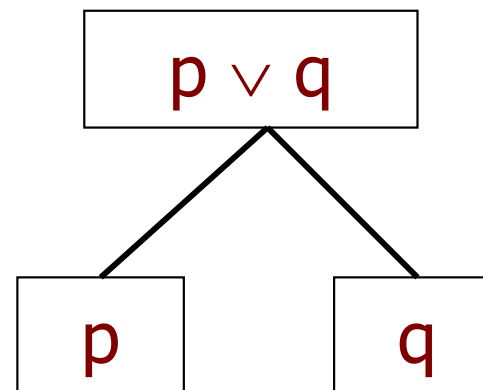
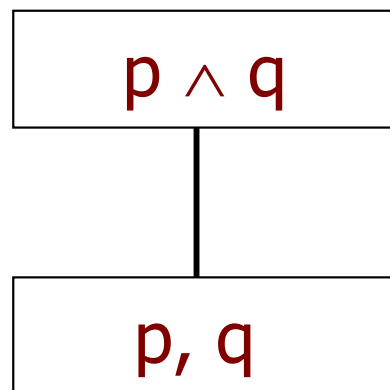
OK

Ellenpélda

Bevezető: Tabló módszer a Boole logika esetén

Kérdés: Hogyan tehető igazzá egy adott kifejezés (Boole-függvény)?

- Alapötlet: A logikai kifejezés felbontása egy **fa struktúrában** (ez a tábló)
 - Csomópontok: Kifejezések, amelyeket igazzá akarunk tenni
 - Élek képzése: **Konstrukciós szabályokkal** az operátorok jelentése alapján;
Elágazás megjelenik: Ha többféleképpen igazzá tehető egy kifejezés
- A felbontás előtt a kifejezést **negált normál formára** kell hozni:
Negálás ne legyen összetett kifejezések előtt
 - de Morgan azonosságok: $\neg(p \vee q) = (\neg p) \wedge (\neg q)$, $\neg(p \wedge q) = (\neg p) \vee (\neg q)$
- Tábló konstrukciós szabályok Boole logika esetén:



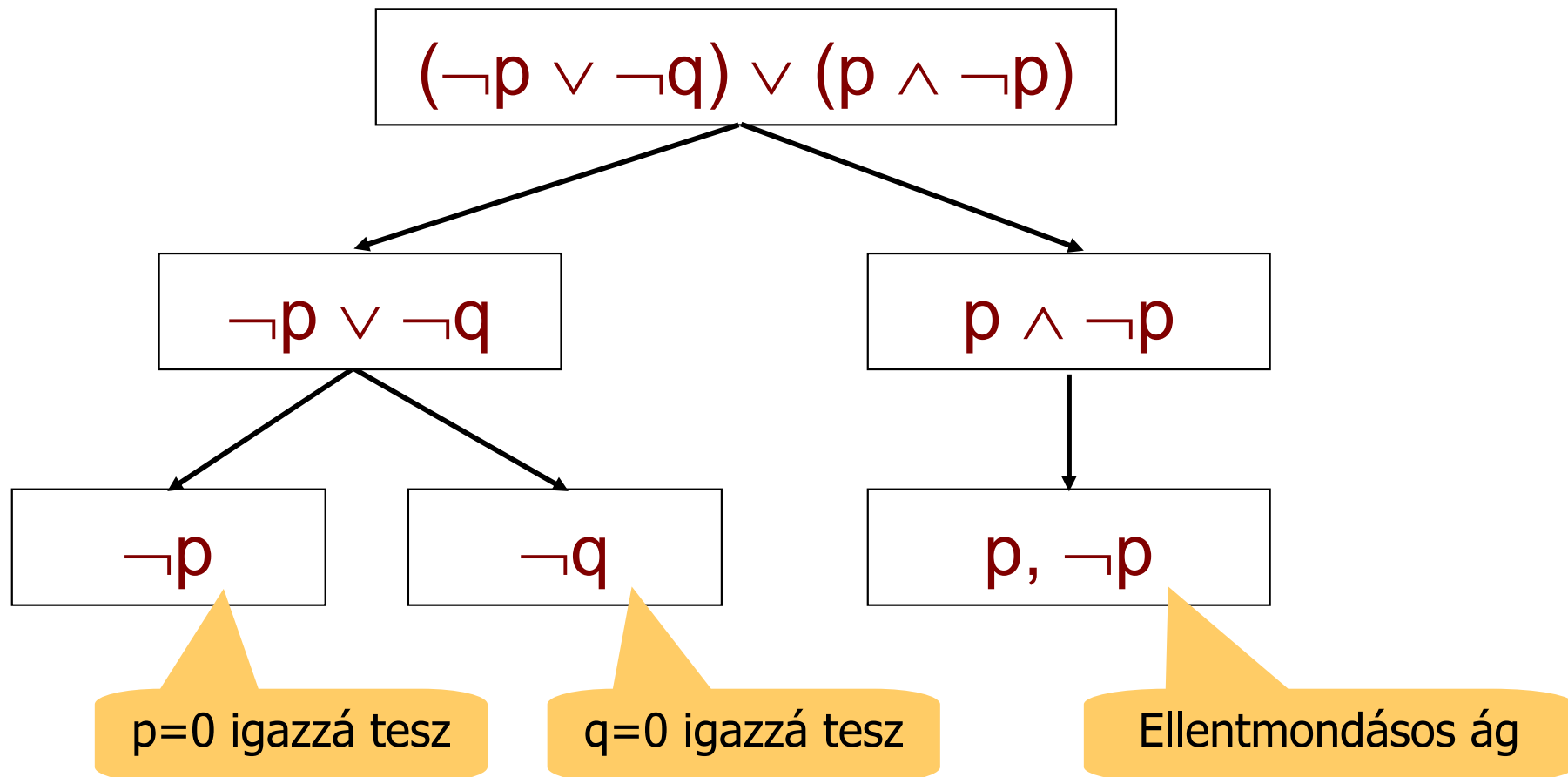
Bevezető: A tábló kiértékelése a Boole logika esetén

Meddig folytassuk a felbontást?

- **Egy ág (felbontás) terminálása:**
 - Ponált vagy negált változók listája marad a csomópontban, operátor nem
 - A lista minden elemét igazgá kell tenni a változók behelyettesítésével (a változók igaz vagy hamis értéket kaphatnak)
- **Egy-egy ág terminálása után:**
 - **Ellentmondásos ág:** Ugyanaz a változó ponált és negált formában is előfordul; nincs konzisztens behelyettesítés
 - Pl. $p, \neg p$ esetén ellentmondás, egyszerre nem lehet igaz a két kifejezés
 - **Sikerés ág:** Nincs ellentmondás; a lista minden eleme igazgá tehető behelyettesítéssel
 - Pl: $p, \neg q$ esetén: p igaz, q hamis az a behelyettesítés, ami igazgá teszi
 - Az így adódó behelyettesítéssel a kezdeti kifejezés igazgá tehető
- A fa sikeres ágai jelölik ki, hol igaz a kifejezés

Bevezető: Egy tábló konstruálása Boole logika esetén

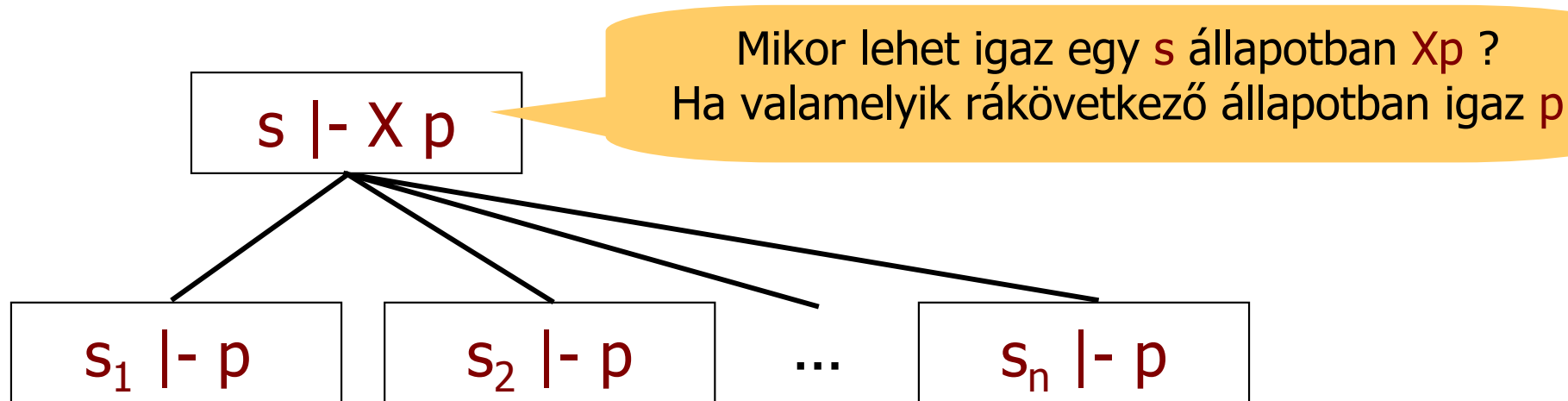
- Eredeti kifejezés: $\neg(p \wedge q) \vee \neg(\neg p \vee p)$
- Negálás bevitele: $(\neg p \vee \neg q) \vee (p \wedge \neg p)$
- Tábló konstruálás:



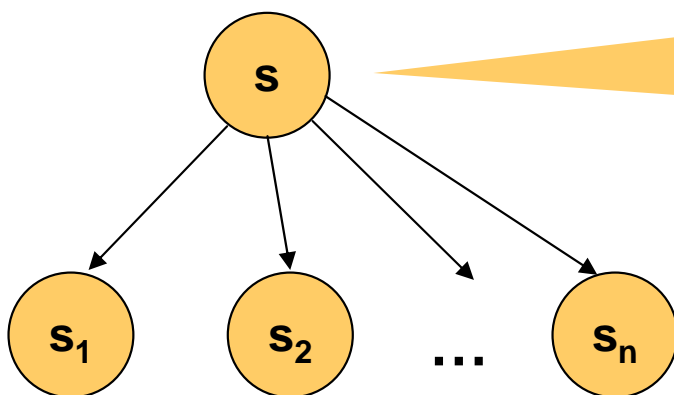
A tábló kiterjesztése PLTL-re

- Modellellenőrzés: Ellenpéldát keres adott kifejezéshez, tehát negált kifejezésből készül a tábló!
 - Ha van sikeres (nem ellentmondásos) ág, az ellenpéldát ad!
 - Ha minden ág ellentmondásos, akkor az eredeti kifejezés igaz!
- Új felbontási szabályok kellenek a temporális operátorokhoz
 - Újdonság: A felbontás a modell alapján végezhető
 - Jelölés: $s \models p$ jelöli, hogy p igazságát keressük s állapotból indulva
- Atomi kijelentések kezelése:
 - $s \models P$ sikeres, ha $P \in L(s)$
 - $s \models P$ ellentmondásos, ha $P \notin L(s)$
 - $s \models \neg P$ sikeres, ha $P \notin L(s)$
- Temporális operátorok:
 - X és U felbontása elég (a többiek ezekkel kifejezhetők)
 - A szabályok a modellre fognak utalni

Felbontás az X operátor esetén



amennyiben a modellben:

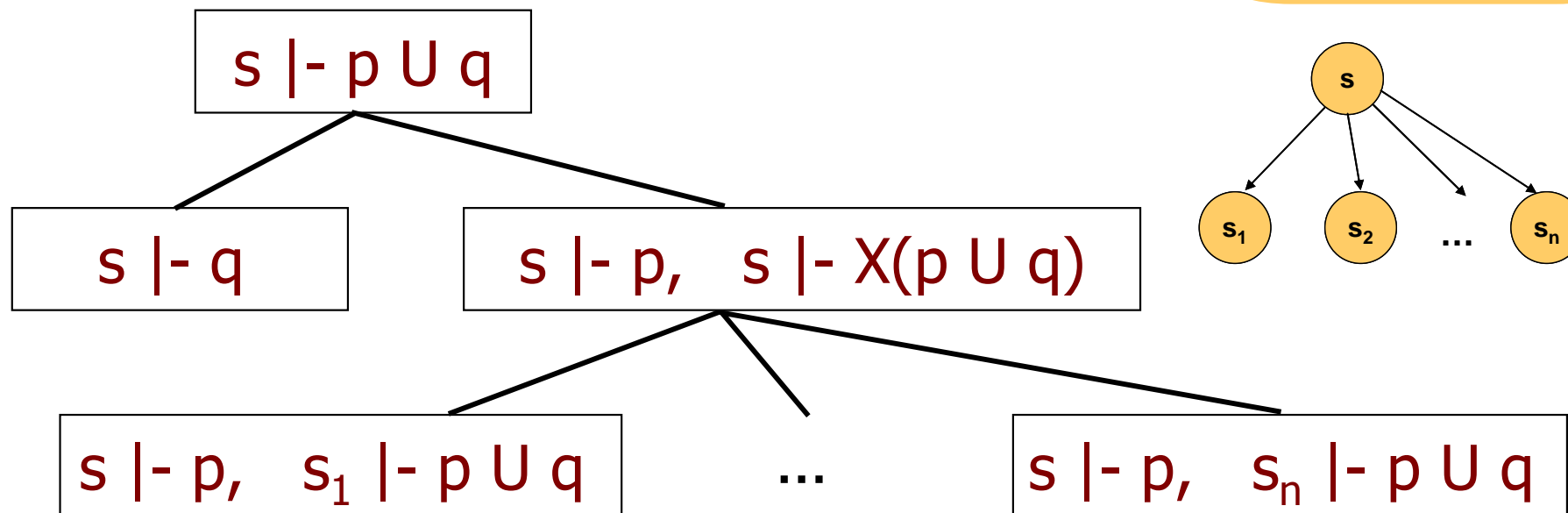


Közvetlen
ellentmondás,
ha s -nek nincs
rákövetkező
állapota!

Felbontás az U operátor esetén

- Felhasználjuk: $p \cup q = q \vee (p \wedge X(p \cup q))$

Véges útvonalra külön figyelünk



- A felbontás meddig folytatódik?

- **Ellentmondásra jutunk:**

- Atomi kijelentésre vonatkozó lokális állítás nem teljesül
- Útvonal végén X operátor van
- Ciklus alakul ki p ágon, q teljesülése nélkül

- **Sikeres ágak:**

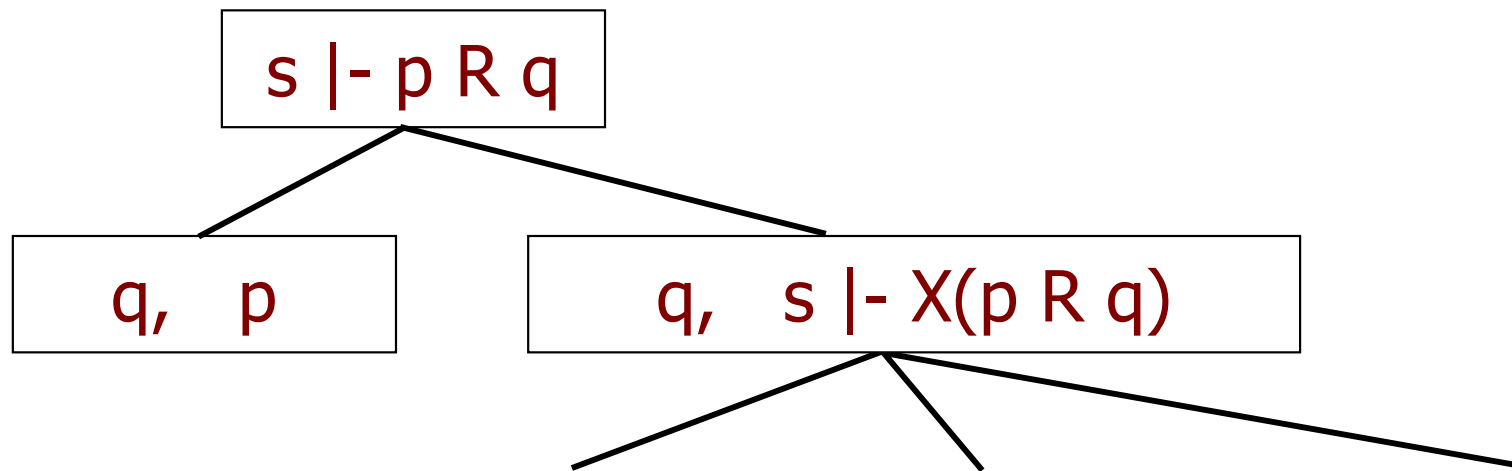
- Atomi kijelentésekre vonatkozó állítás(ok) teljesül(nek)
- Ciklus alakul ki, és nincs ellentmondás

Egy speciális operátor: R

- Negált normál formára hozás az U operátor esetén:
 - $\neg(p \cup q) = ?$
 - Bevezethető az U operátor „duálisa”, az R (Release)
 - Definíció: $p R q = q \wedge (p \vee X(p R q))$
 - A negált normál formára hozás az R használatával:

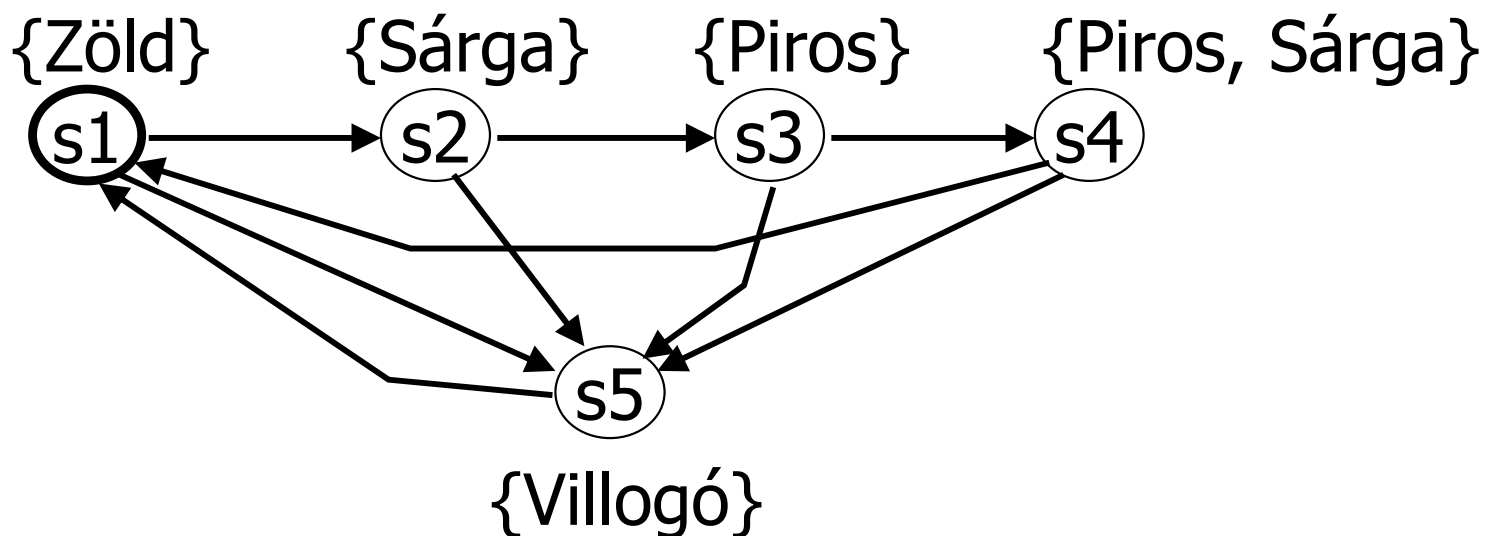
$$\neg(p \cup q) = (\neg p) R (\neg q)$$

- Az R operátor tablója:



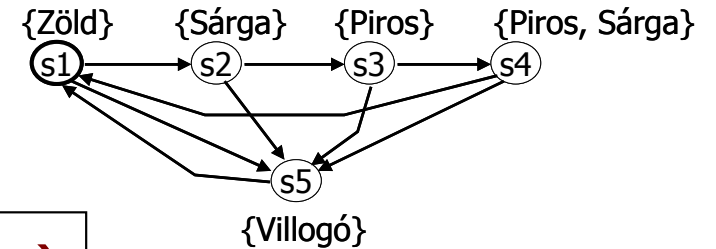
Egy példa

- A közlekedési lámpa módosított modellje (KS)
- Igaz-e, hogy ha a lámpa a kezdeti állapotban **zöld**, akkor előbb-utóbb **piros** lesz?
 - Az ellenőrizendő PTL kifejezés: **Zöld** \Rightarrow **F Piros**



A kifejezés tablója

- A kifejezés negálása: $s_1 \vdash \neg(\text{Zöld} \Rightarrow \text{F Piros})$
- Negált normál forma (ld. $P \Rightarrow Q = \neg P \vee Q$):
 $\neg(\text{Zöld} \Rightarrow \text{F Piros}) = \text{Zöld} \wedge \neg \text{F Piros} = \text{Zöld} \wedge G(\neg \text{Piros})$
- A tábló konstruálása:



s1 állapotban van Zöld címke

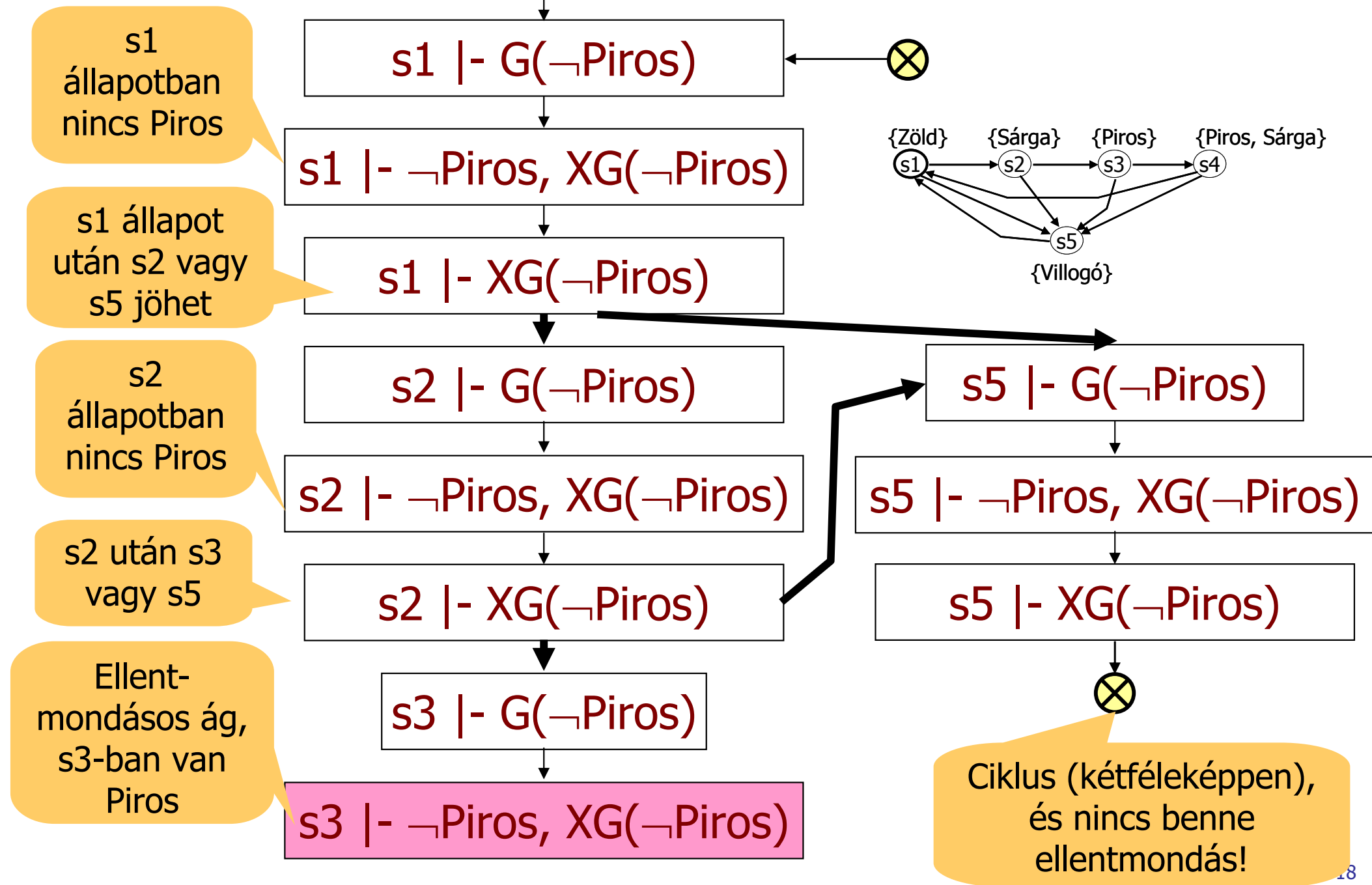
$s_1 \vdash \text{Zöld} \wedge G(\neg \text{Piros})$

$s_1 \vdash \text{Zöld}, s_1 \vdash G(\neg \text{Piros})$

Egyszerűsített jelölés
($s_1 \vdash \text{Zöld}$ kimarad)

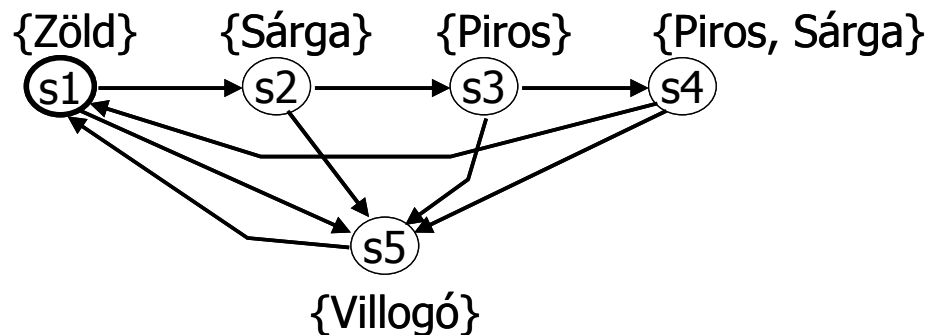
$s_1 \vdash G(\neg \text{Piros})$

A kifejezés táblójának folytatása

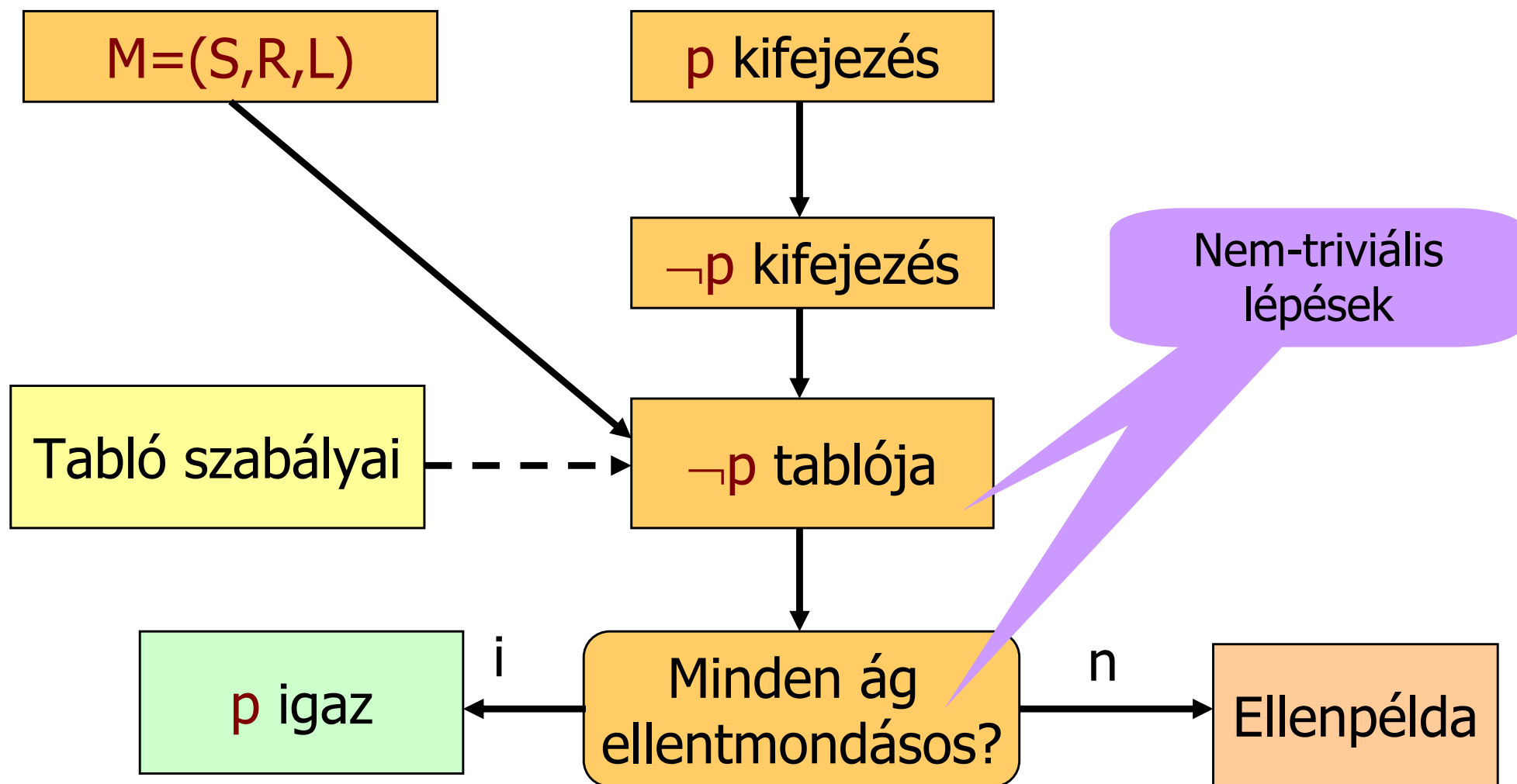


A modellellenőrzés eredménye

- A tábló eredménye:
 - Egy ellentmondásos ág
 - Két ellentmondás nélküli ciklus
- Következtetés:
 - Vannak olyan lefutások, ahol a **negált kifejezés teljesül**:
Ciklus 1: s1, s2, s5, s1, ...
Ciklus 2: s1, s5, s1, ...
- Az eredeti kifejezés **Zöld** \Rightarrow **F Piros** tehát nem igaz
 - Ellenpéldák adhatók

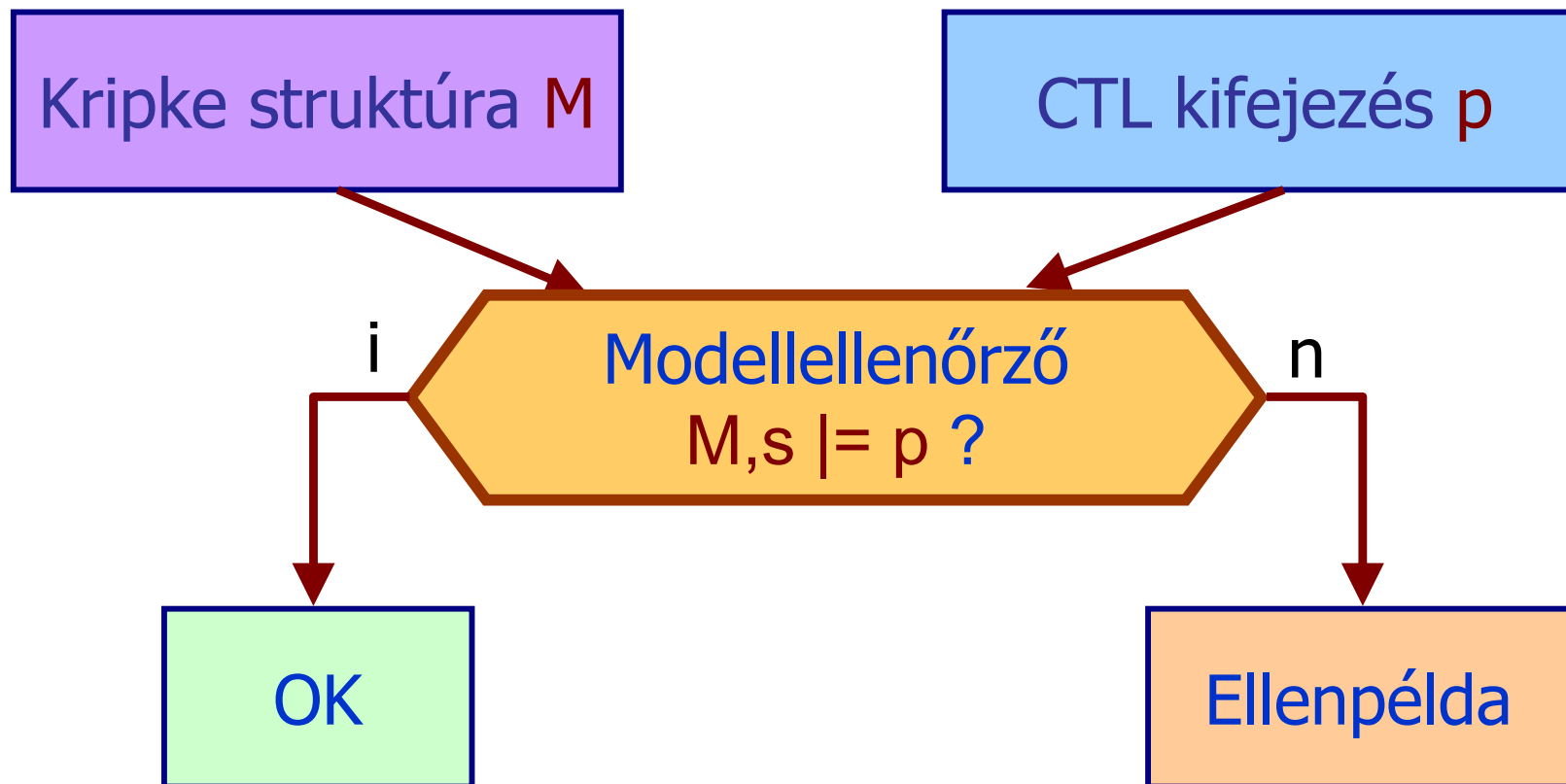


A tábló módszer összefoglalása



CTL modellellenőrzés: Szemantika alapon

A modellellenőrzés feladata

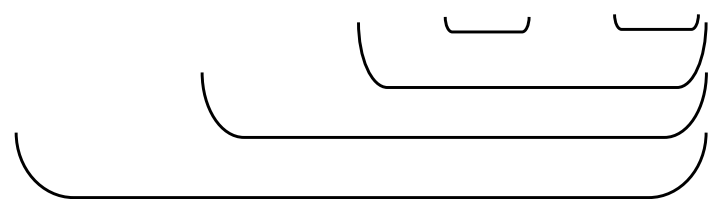


Alapötlet: Állapotok címkézése

- **Globális modellellenőrzés:**
 - Jelölés: $Sat(p)$ jelöli egy p CTL kifejezés esetén azoknak az állapotoknak a halmazát, ahol p igaz
 - Címkézés: Ezeket az állapotokat p -vel címkézzük
 - Ezután $s \in Sat(p)$ egyszerűen vizsgálható egy adott s állapotra (pl. kezdőállapot is): Szerepel-e rajta a p címke?
- A címkézés (azaz $Sat(p)$ számítása) inkrementálisan történik
 - Címkézett állapothalmazok bővítése
 - Az iteráció vége: Nem nő a címkézhető állapotok halmaza

CTL modellellenőrzés állapot címkézéssel

- Állapotok címkézése: ahol igaz egy adott kifejezés
- Összetett kifejezés esetén hogyan történik a címkézés?
 - Kifejezések felbontása azok struktúrája alapján, és „belülről kifelé” $\text{Sat}(\text{kifejezés})$ számítások:

$$\text{AF} (P \wedge E (Q \cup R))$$


- Algoritmus az összetett kifejezés felbontása alapján:
 - Kiindulás: KS címkézve van **atomi kijelentésekkel**
 - Tovább lépés: Címkézés az összetettebb kifejezésekkel
 - Ha p illetve q címkék már vannak, akkor megadható, hol lehet $\neg p$, $p \wedge q$, $\text{EX } p$, $\text{AX } p$, $E(p \cup q)$, $A(p \cup q)$ címke
 - Ehhez felhasználhatók a szemantika szabályai
 - Inkrementális címkézés történik

Hol igaz egy adott kifejezés?

- P (atomi kijelentés) azokban az s állapotokban igaz, ahol $P \in L(s)$
 - Itt P címkeként már szerepel a KS-ba
- $\neg P$ azokban az s állapotokban igaz, ahol $P \notin L(s)$
 - Ezek az állapotok $\neg P$ kifejezéssel címkézhetők
- $p \wedge q$ azokban az s állapotokban igaz, ahol p és q is igaz
 - Egy állapot címkézése lehet $p \wedge q$, ha címkéi között már van p és q
- Temporális operátorok: $EX, AX, E(U), A(U)$
 - Bonyolultabb algoritmust igényel a címkézés!

Az AX, EX alakú kifejezések

- **EX p** azokban az **s** állapotokban igaz, amelyeknek van olyan rákövetkező állapota, ahol **p** igaz
 - Egy állapot címkézése lehet **EX p**, ha van olyan rákövetkező állapota, ami **p**-vel címkézett



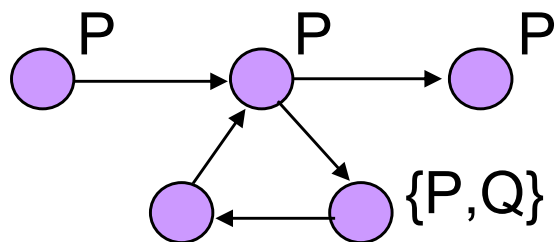
- **AX p** azokban az **s** állapotokban igaz, amelyeknek minden rákövetkező állapotában **p** igaz
 - Egy állapot címkézése lehet **AX p**, ha minden rákövetkező állapota **p**-vel címkézett



Az $E(p \cup q)$ kifejezések

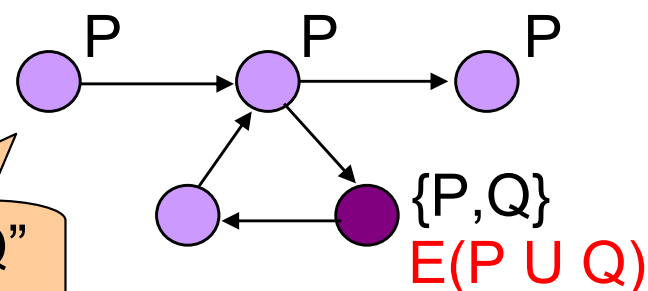
- Hol igaz $E(p \cup q)$?
 - Felhasználható: $E(p \cup q) = q \vee (p \wedge EX E(p \cup q))$
 - „Rekurzív” képlet...
- Tehát mely s állapotok címkézhetők $E(p \cup q)$ -val?
 - Ha s címkézett q -val, vagy
 - ha s címkézett p -vel, és legalább egy rákövetkezője már címkézett $E(p \cup q)$ -val
- Iteráció adódik:
 - q -val már címkézett állapotok adják azokat az állapotokat, ahol először megjelenik az $E(p \cup q)$ címke
 - Ezek megelőző állapotait kell végignézni:
Ha szerepel ott a p címke, akkor rátehető az $E(p \cup q)$ címke is!
 - Így visszafelé járjuk be azokat az útvonalakat, amik p -vel címkézett állapotokon keresztül visznek q -val címkézett állapotba

Az $E(P \cup Q)$ címkézés iterációja

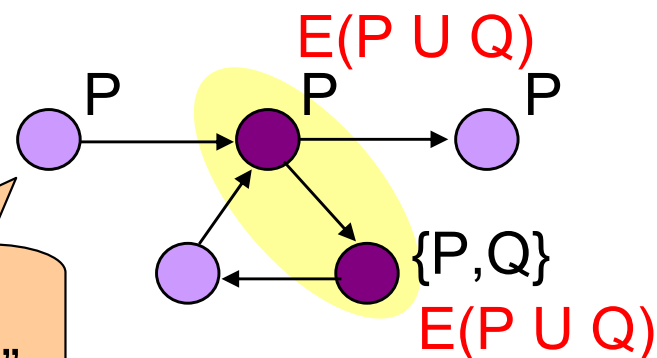


Kripke struktúra a kezdő címkézéssel

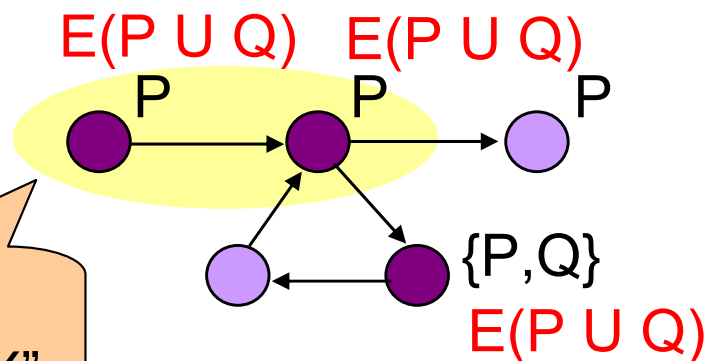
Első lépés: „Q”



Második lépés: „ $P \wedge EX$ ”



Harmadik lépés: „ $P \wedge EX$ ”



- Az iteráció addig tart, míg nő az állapothalmaz (fixpontot érünk el)

Az $A(p \cup q)$ kifejezések

- Hol igaz $A(p \cup q)$?
 - Felhasználható: $A(p \cup q) = q \vee (p \wedge AX A(p \cup q))$
 - Ez is „rekurzív” képlet...
- Tehát mely s állapotok címkézhetők $A(p \cup q)$ -val?
 - Ha s címkézett q -val, vagy
 - ha s címkézett p -vel, és minden rákövetkezője már címkézett $A(p \cup q)$ -val
- Iteráció adódik:
 - q -val már címkézett állapotok adják azokat az állapotokat, ahol először megjelenik az $A(p \cup q)$ címke
 - Ezek megelőző állapotait kell végignézni:
Ha szerepel ott a p címke, és minden rákövetkező állapotukon szerepel az $A(p \cup q)$ címke, akkor ezekre is rátehető az $A(p \cup q)$ címke

Ezzel a formális szintaxisban használt operátorokat lefedtük!

Még egy példa az iterációra

- Hol igaz $AF\ p$?
 - Felhasználható: $AF\ p = p \vee AX\ AF\ p$
 - „Rekurzív” képlet...
- Tehát mely s állapotok címkézhetők $AF\ p$ -vel?
 - Ha s címkézett p -vel, vagy
 - ha minden rákövetkezője már címkézett $AF\ p$ -vel
- Iteráció adódik:
 - p -vel már címkézett állapotok adják azokat a kiinduló állapotokat, ahol először megjelenik az $AF\ p$ címke
 - Ezek megelőző állapotait kell végignézni:
Ha minden rákövetkező állapotukon szerepel az $AF\ p$ címke, akkor rátehető az $AF\ p$ címke
 - Így visszafelé keressük azokat az állapotokat, amik minden útvonalon p -vel címkézett állapotokba vezetnek

Iteráció halmazműveletekkel

- A címkézés bővítése halmazműveletekkel történik
 - Kezdőhalmaz: Rész-kifejezésekkel már címkézett állapotok
 - Címkézés bővítése:
 - $E(p \cup q)$ esetén: „Legalább egy rákövetkező állapota már címkézett ...”
 - $A(p \cup q)$ esetén: „Minden rákövetkező állapota már címkézett ...”
 - Ez alapján a megelőző állapotok valamelyikére tehető az új címke
- Hogyan definiálhatók ezek a halmazok formálisan?
 - Már címkézett Z állapot-halmaz alapján:
$$\text{pre}_E(Z) = \{s \in S \mid \text{létezik olyan } s', \text{ hogy } (s, s') \in R \text{ és } s' \in Z\}$$

$$\text{pre}_A(Z) = \{s \in S \mid \text{minden } s'\text{-re, ahol } (s, s') \in R: s' \in Z\}$$
- Példa: $E(P \cup Q)$ iterációja:
 - Kezdőhalmaz: $Z_0 = \{s \mid Q \in L(s)\}$
 - Iteráció: $Z_{i+1} = Z_i \cup (\text{pre}_E(Z_i) \cap \{s \mid P \in L(s)\})$
 - Eddig címkézettek és ...
 - a megfelelő megelőző állapotok amelyek ...
 - ... P-vel címkézettek
 - Iteráció vége: Ha $Z_{i+1} = Z_i$, azaz nem bővül a halmaz

CTL modellellenőrzés: Összefoglalás

- Globális modellellenőrzés:
 - Állapotok címkézése azokkal a (rész)kifejezésekkel, amelyek igazak az adott állapotban
- Kifejezések felbontása (szintaxis szabályok alapján)
 - Atomi kijelentésekből indítva az összetettebb kifejezések felé („belülről kifelé”)
 - Az előző lépésben adott címkézés felhasználása
- **EX, AX** esetén: Megelőző állapot vizsgálata
- **E(p U q), A(p U q)** esetén: Inkrementális címkézés
 - Kezdőhalmaz:
 - A belső kifejezések (**p, q**) által meghatározott állapothalmazok alapján
 - Iteráció: A szemantika alapján (megelőző állapotokra lépegetve)
 - Iteráció vége: Nem nő a címkézett állapotok halmaza
 - Precíz matematikai algoritmus: Fixpont számítás teljes hálókbán

A bevezető példa kifejtése

- Kifejezések felbontása azok struktúrája alapján, és „belülről kifelé” címkézés:

$AF (P \wedge E (Q U R))$

Q és R címkék
a KS-ban

Inkrementális címkézés: $E (U)$
Az iteráció végén megjelenik
az $E(Q U R)$ címke

Itt P -vel és $E(Q U R)$ -val címkézett
állapothalmazok metszete
(mintha az $E(Q U R)$ címke egy
„atomi kijelentés” lenne):
Megjelenik a $P \wedge E(Q U R)$ címke

Inkrementális címkézés: AF alapján
(mintha a $P \wedge E(Q U R)$ címke egy
„atomi kijelentés” lenne):
Megjelenik az $AF(P \wedge E(Q U R))$ címke.
Ez a kezdőállapotra ellenőrizhető.

Összefoglalás

- PLTL modellellenőrzés
 - Tabló konstruálása
 - Boole-logikai bevezetés: Ellentmondásos és sikeres ágak
 - Tabló PLTL esetén: Ellenpélda keresés (negált követelményre)
- CTL modellellenőrzés
 - Szemantika alapú modellellenőrzés
 - Inkrementális címkézés bővülő rész kifejezésekkel (globális modellellenőrzés)
 - Halmazműveletekkel történik

Hogyan tehető hatékonyá ez az algoritmus?

PLTL modellellenőrzés:
Automata alapú megközelítés
(kiegészítő anyag,
számunkérésen nem szerepel)

Automaták véges hosszúságú szavakon

- $A=(\Sigma, S, S_0, \rho, F)$ ahol
 - Σ az ábécé, S állapotok, S_0 kezdőállapotok
 - ρ az állapotátmeneti reláció, $\rho: S \times \Sigma \rightarrow 2^S$
 - F az elfogadó állapotok halmaza.
- Az automata futása:
 - Egy beérkező $w=(a_0, a_1, a_2, \dots a_n)$ betűsorozat hatására egy $r=(s_0, s_1, s_2, \dots s_n)$ állapotsorozat
 - r elfogadó futás, ha $s_n \in F$
 - Egy w szót elfogad az automata, ha létezik rá elfogadó futás
- $L(A)=\{ w \in \Sigma^* \mid w \text{ elfogadott} \}$
az automata által elfogadott nyelv

Automaták végtelen hosszúságú szavakon

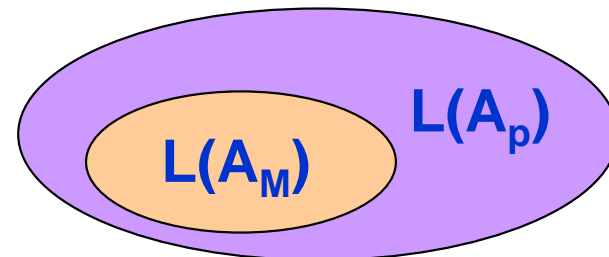
- Alkalmazás: Folyamatosan működő rendszerek
 - Nem ellenőrizhető, hogy a végállapot elfogadó-e
- Büchi elfogadási kritérium:
 - Egy beérkező $w=(a_0, a_1, a_2, \dots)$ betűsorozat hatására egy $r=(s_0, s_1, s_2, \dots)$ állapotsorozat
 - $\text{lim}(r)=\{s \mid s \text{ előfordul végtelenül sokszor, azaz nincs olyan } j, \text{ hogy } \forall k>j: s \neq s_k\}$
 - Elfogadó a futás, ha $\text{lim}(r) \cap F \neq \emptyset$
 - Egy w szót elfogad az automata, ha létezik rá elfogadó futás (azaz végtelen sokszor érint elfogadó állapotot)
- $L(A)=\{w \in \Sigma^* \mid w \text{ elfogadott}\}$ az automata által elfogadott nyelv

Az automata alapú módszer

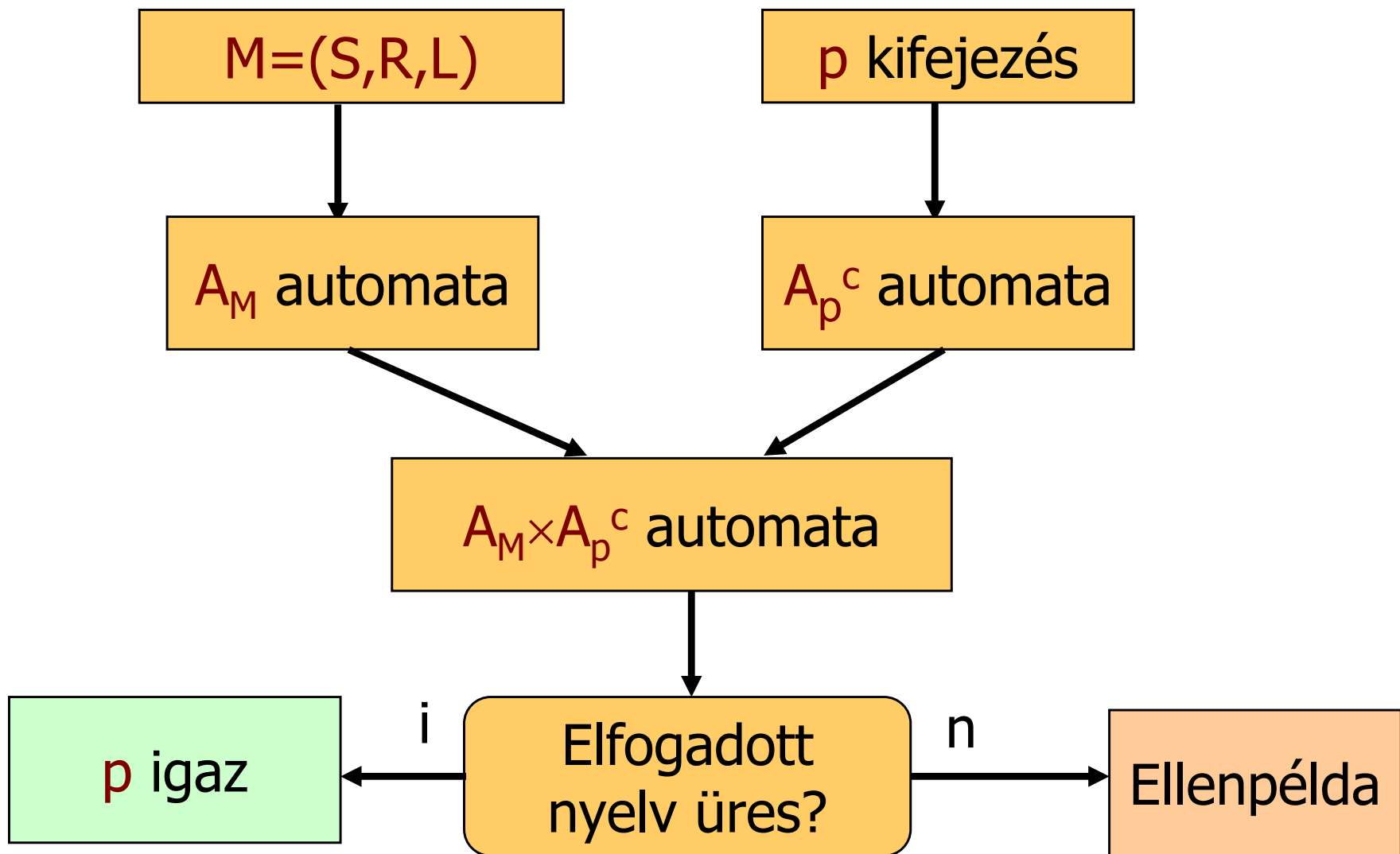
- A **KS** egy **s** állapothoz: $L(s)$ a betű a 2^{AP} ábécéből
 - Pl. {Piros, Sárga} az ábécé egy betűje
- A $\pi = (s_0, s_1, s_2, \dots, s_n)$ útvonal egy szót azonosít:
 $(L(s_0), L(s_1), L(s_2), \dots, L(s_n))$
- Két automatát kell konstruálni:
 - $M = (S, R, L)$ alapján egy A_M automata konstruálható, amely azokat és csakis azokat a szavakat fogadja el, amelyek megfelelnek M útjainak.
 - p kifejezés alapján egy A_p automata konstruálható, amely azokat és csakis azokat a szavakat fogadja el, amelyek olyan utakhoz tartoznak, ahol p igaz
 - Itt felhasználhatók a tábló képzés szabályai:
Nem a modell alapján bontunk, hanem állapotokat generálunk

Modellellenőrzés az automatákkal

- Kérdés: $L(A_M) \subseteq L(A_p)$, vagyis a „modell” nyelv része-e a „tulajdonság” nyelvnek?
 - Ha igen, akkor $M \models p$
- A kérdés átalakítása:
 - Nyelvek metszetének ürességét kell vizsgálni:
 $L(A_M) \cap L(A_p)^c = \emptyset$, itt $L(A_p)^c$ a komplementer nyelv
 - Az A_M „modell automata” és az A_p^c „komplementer tulajdonság automata” $A_M \times A_p^c$ szinkron szorzatát képezve, az általa elfogadott nyelv üres-e?
 - Ha üres, akkor $M, \pi \models p$ teljesül
 - Az elfogadott nyelv üres, ha nincs elérhető elfogadó állapot
- Folyamatosan működő rendszerek:
 - Automaták végtelen hosszúságú szavakon;
Büchi elfogadási kritérium: cikluskeresésre vezet



Az automata alapú modellellenőrzés



„On-the-fly” modellellenőrzés

- Alapötlet:
 - Az A_p automata generálása közben lehet elvégezni a szinkron szorzat automata konstruálását is
- Szinkron szorzat automata konstruálása:
 - Az ellenőrizendő kifejezés által vezérelten történik: ahogyan az A_p automata új állapota előáll, úgy kell A_M állapotait „előkeresni”
 - Nem szükséges hozzá a modell állapottér teljes generálása
 - Pl. egy magasabb szintű modellből való származtatás esetén