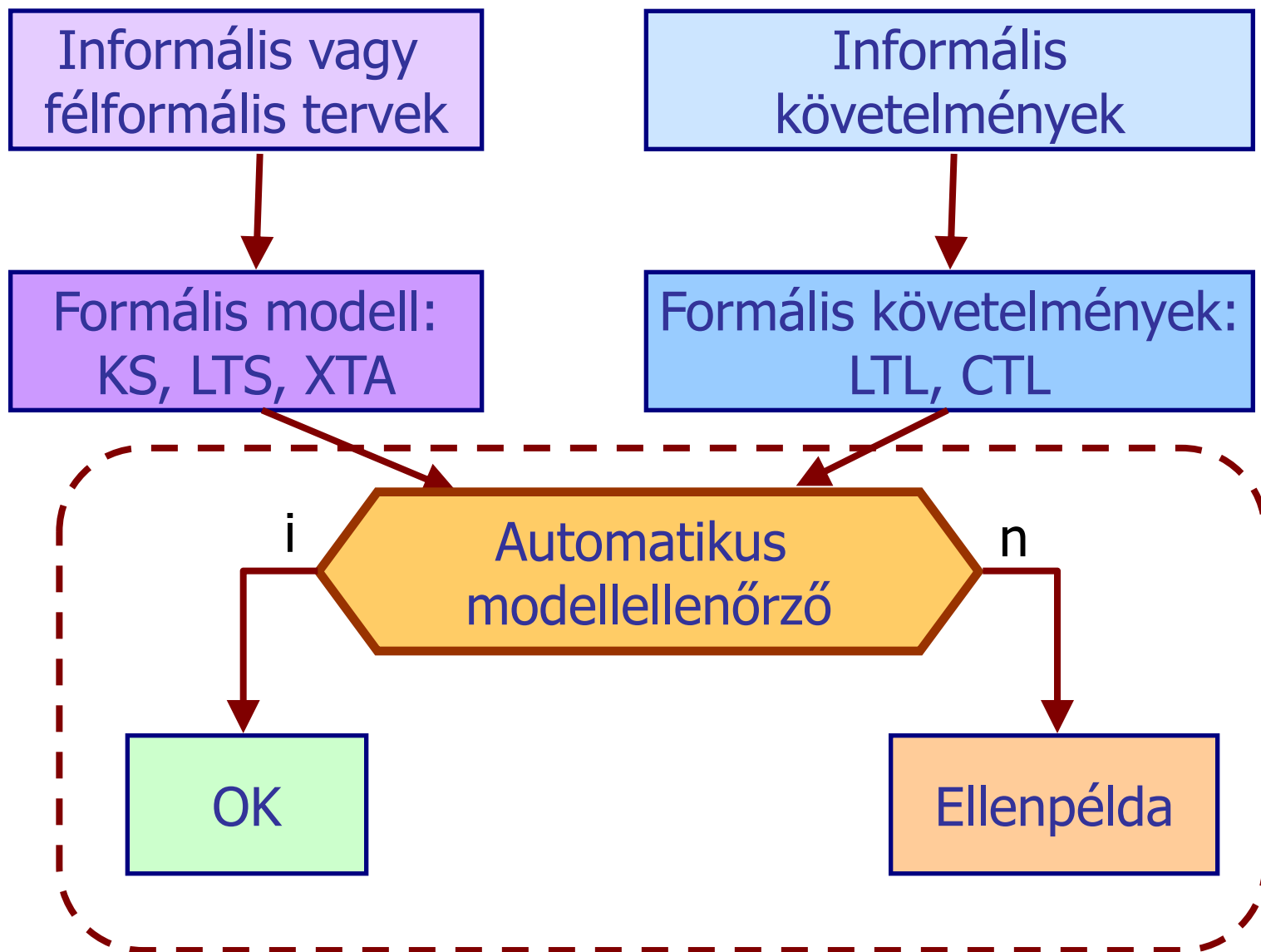


Modellellenőrzés

dr. Majzik István

BME Méréstechnika és Információs Rendszerek Tanszék

Mit szeretnénk elérni?

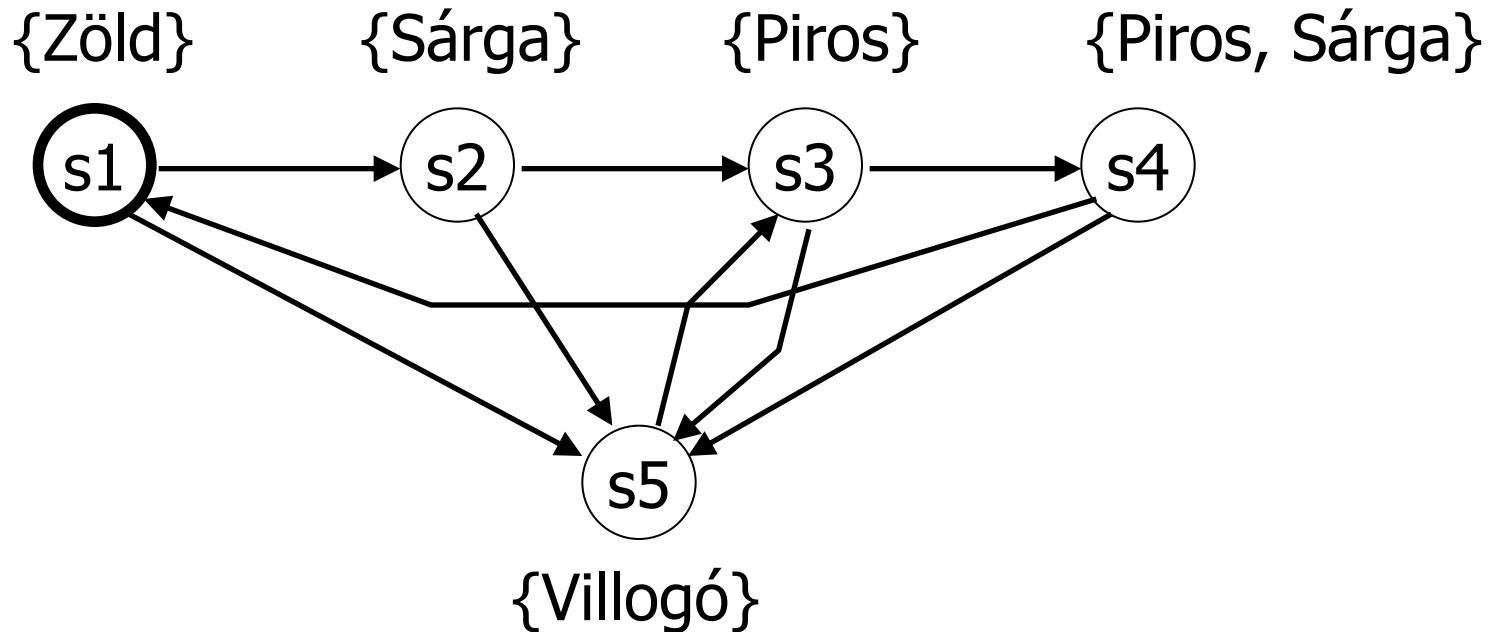


Ismétlés: Kripke-struktúra

$M = (S, R, L)$

Példa: Közlekedési lámpa vezérlője

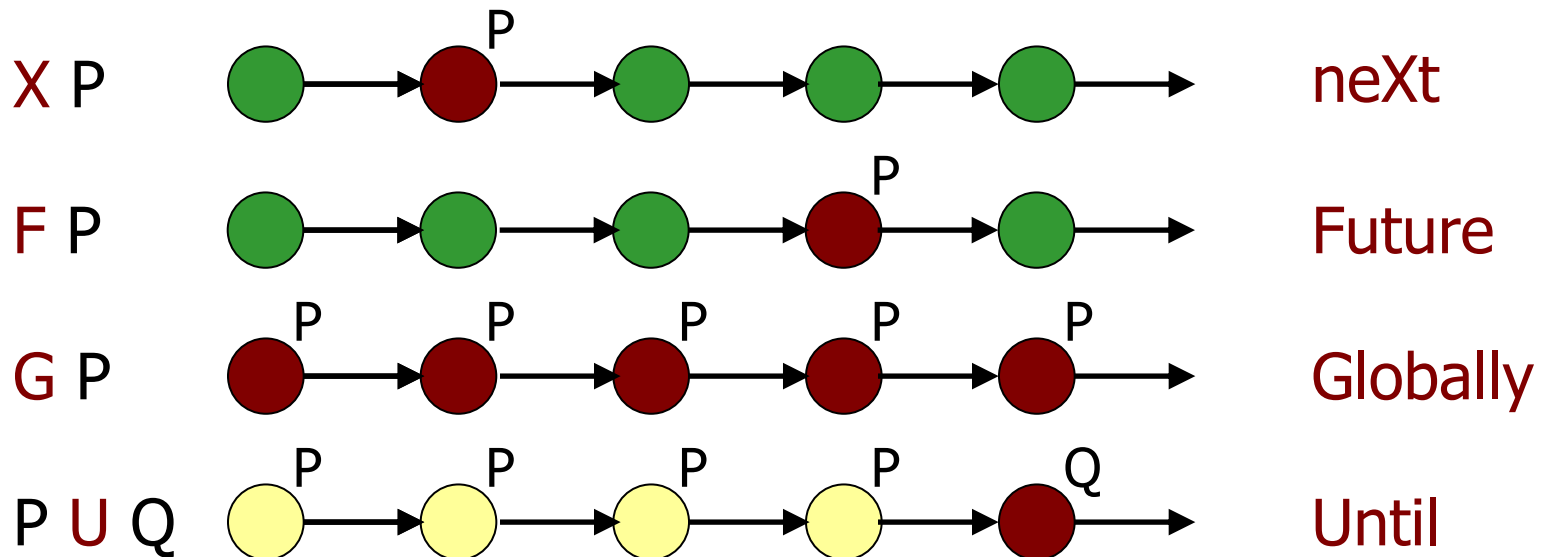
$AP = \{\text{Zöld, Sárga, Piros, Villogó}\}$



Ismétlés: Lineáris idejű temporális logika: PLTL

PLTL elemei:

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

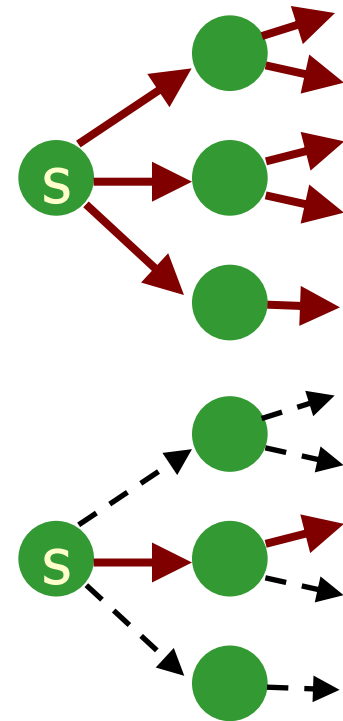


Ismétlés: Elágazó idejű temporális logika: CTL*

CTL* elemei:

- Útvonal kvantorok:

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

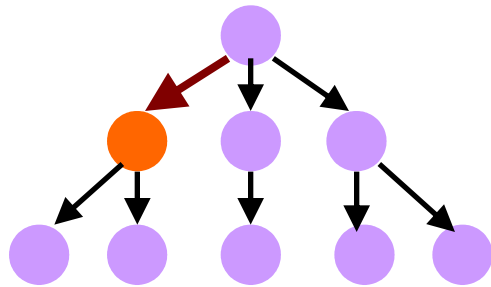


- Útvonalakon kiértékelhető operátorok (mint PLTL):

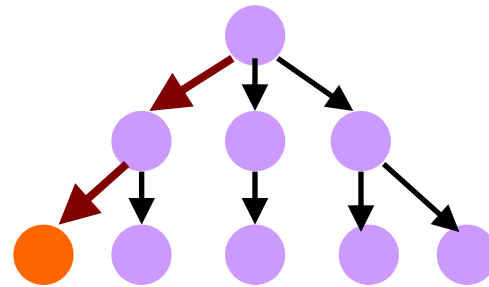
- **X** p, **F** p, **G** p, p **U** q

Ismétlés: Elágazó idejű temporális logika: CTL

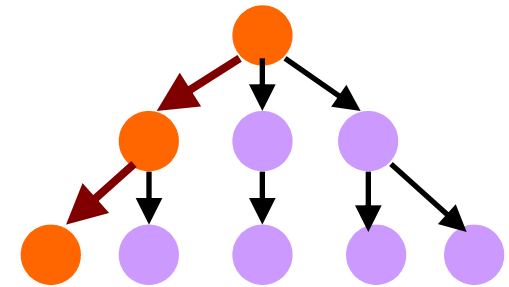
CTL: Állapotokon kiértékelhető összetett operátorok



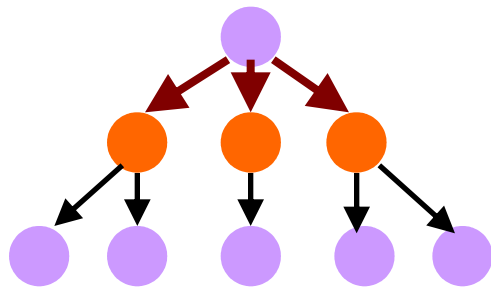
EX P



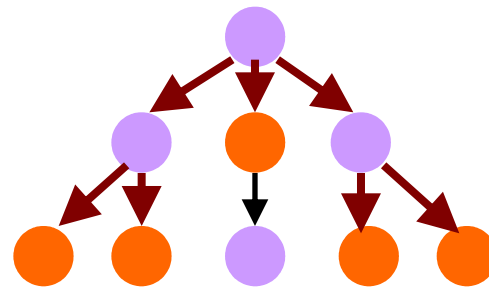
EF P



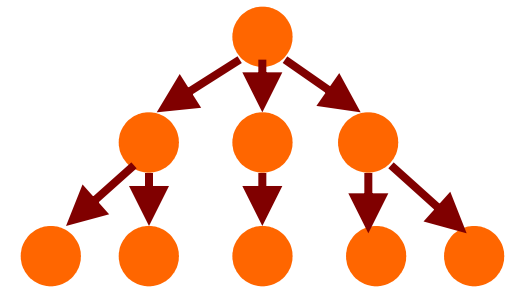
EG P



AX P



AF P



AG P

Az előadás áttekintése

Hogyan működik a modellellenőrzés?

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

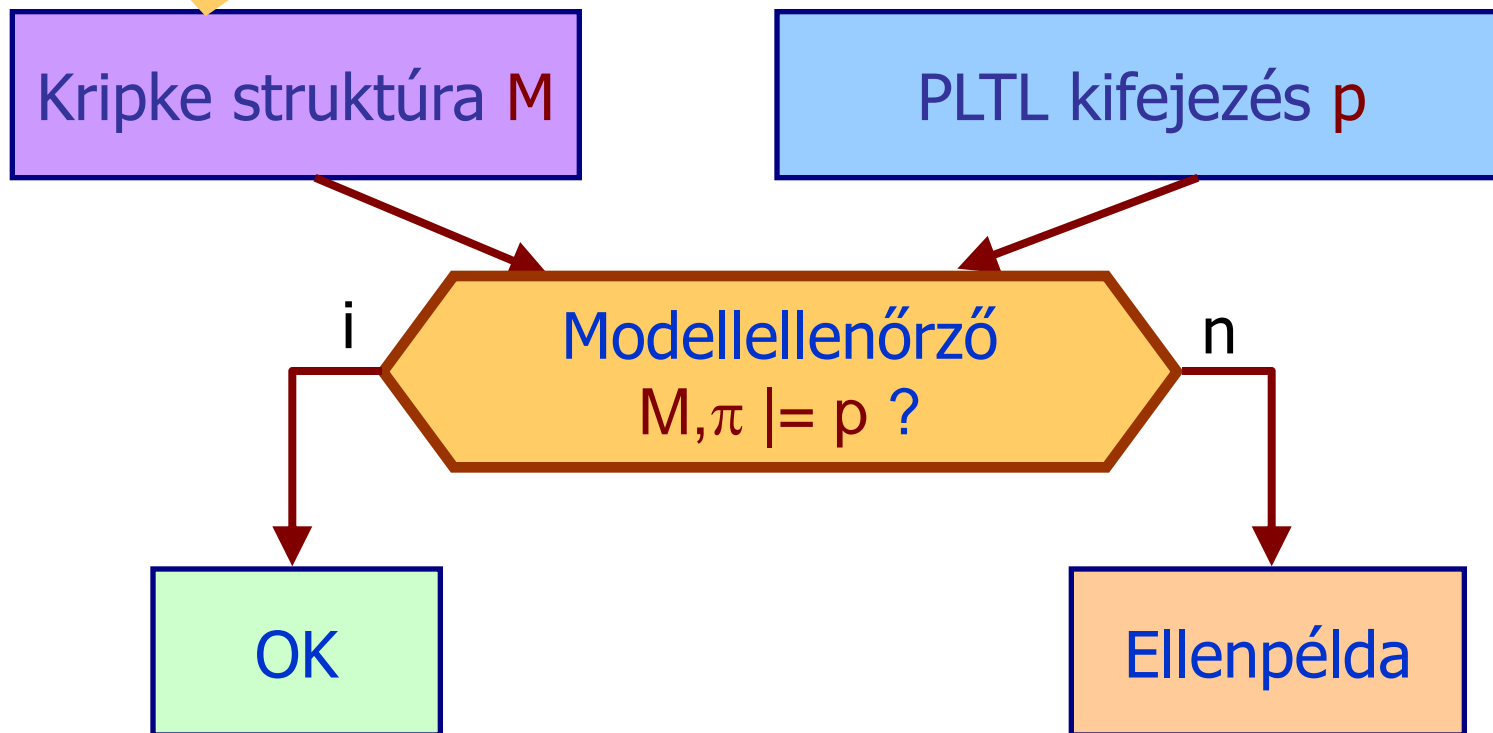
Miért jó ezt tudni?

- Lehetőségek felmérése
 - Korlátok becslése (pl. ellenőrizhető modellek mérete)
 - Hatékony megvalósítás ($10^{69.000}$ állapot? – következő ea.)
- Érdekes alkalmazások alapja
 - Automatikus teszteset-generálás
 - Lépéssorozat generálás szintézis feladathoz

PLTL modellellenőrzés tabló módszerrel

A modellellenőrzés feladata

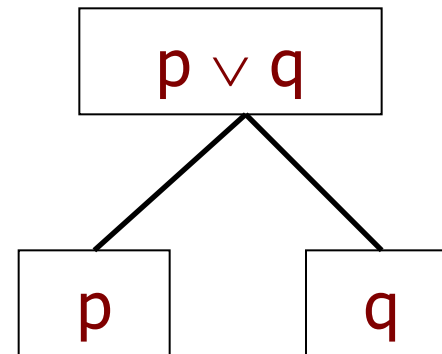
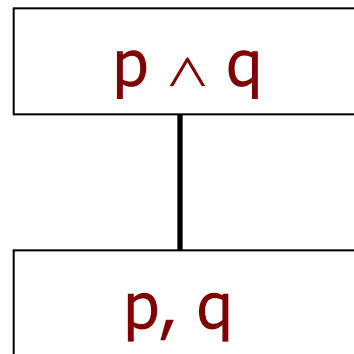
Ha nincs útvonal megadva, akkor a kezdőállapotból induló minden útra ellenőriz



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: **Felbontási szabályokkal** az operátorok jelentése alapján
 - ÉS operátor: Részkifejezések listába gyűjtése (mindegyik igaz kell legyen)
 - VAGY operátor: Elágazás a fában (többféleképpen igazzá tehető)
- Kifejezés felbontási szabályok Boole logika esetén:



- A felbontás előtt a kifejezést ún. **negált normál formára** kell hozni:
Negálás ne legyen összetett kifejezések előtt, csak változók előtt
 - de Morgan azonosságok: $\neg(p \vee q) = (\neg p) \wedge (\neg q)$, $\neg(p \wedge q) = (\neg p) \vee (\neg q)$

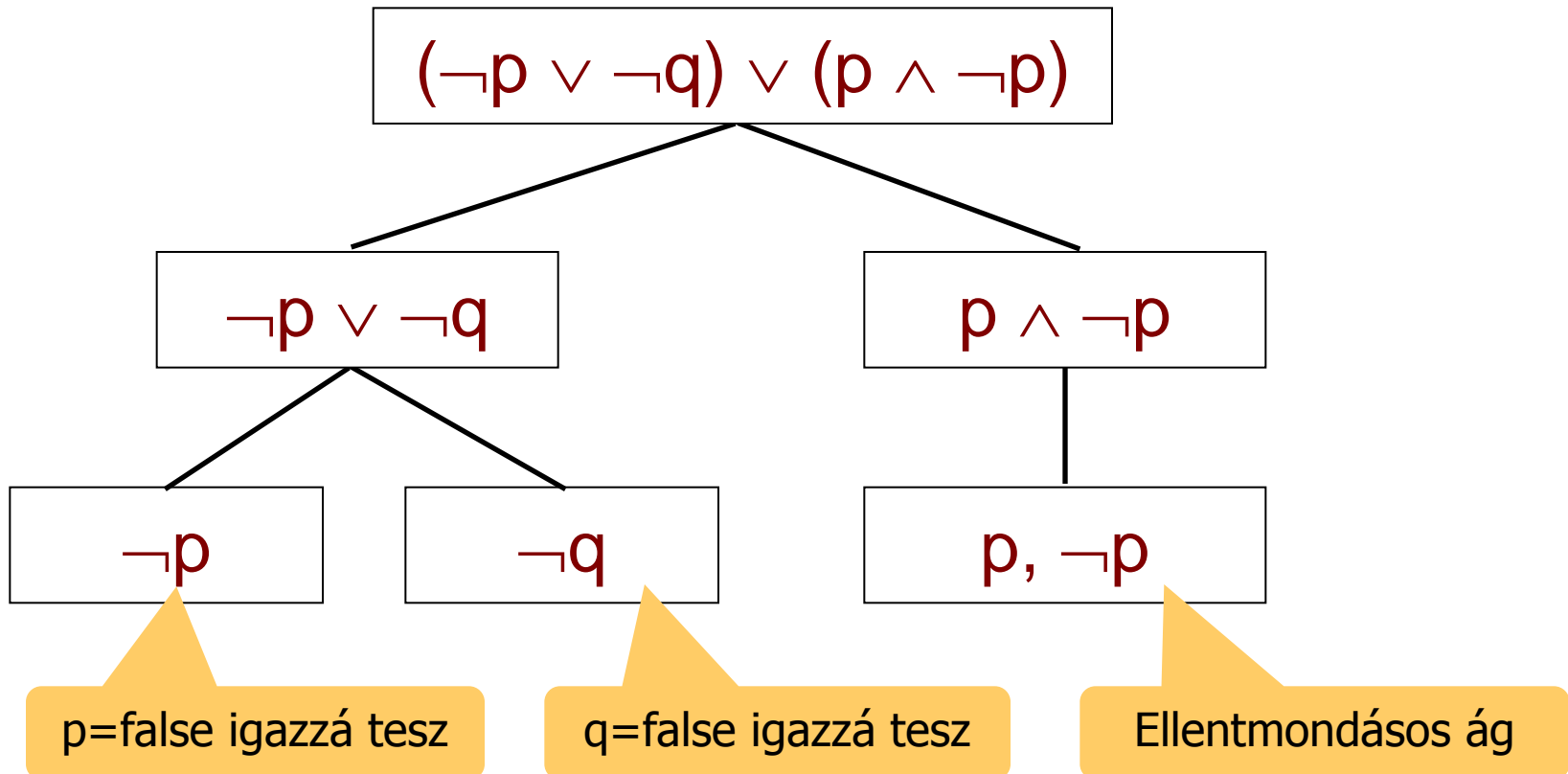
Bevezető: A tábló kiértékelése a Boole logika esetén

Meddig folytassuk a felbontást?

- **Egy ág (felbontás) terminálása:**
 - Operátor nem maradt, csak ponált vagy negált változók listája
 - A lista minden elemét igazá kell tenni a változók behelyettesítésével (a ponált igaz, a negált hamis értéket kap)
- **Egy-egy ág terminálása után:**
 - **Ellentmondásos ág:** Ugyanaz a változó ponált és negált formában is előfordul; nincs konzisztens behelyettesítés
 - Pl. $p, \neg p$ lista ellentmondás, egyszerre nem lehet igaz a két kifejezés
 - **Sikeres ág:** Nincs ellentmondás; a lista minden eleme igazá tehető behelyettesítéssel
 - Pl: $p, \neg q$ lista: p igaz, q hamis az a behelyettesítés, ami igazá teszi
 - Az így adódó behelyettesítéssel a **kezdeti kifejezés igazá tehető**
- A fa sikeres ágai alapján igazá tehető a kifejezés

Bevezető: Egy tábló konstruálása Boole logika esetén

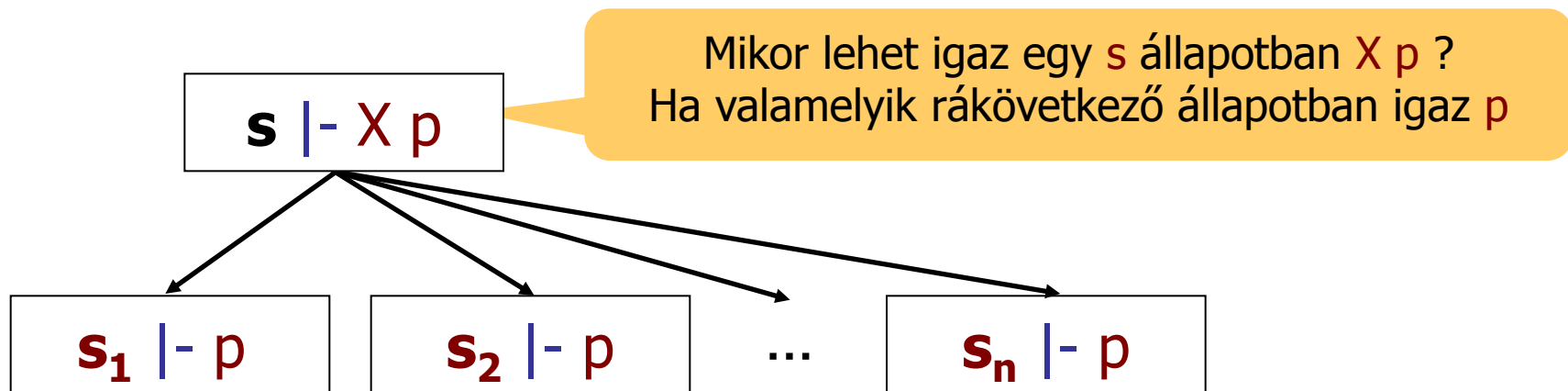
- Eredeti kifejezés: $\neg(p \wedge q) \vee \neg(\neg p \vee p)$
- Negálás bevitele: $(\neg p \vee \neg q) \vee (p \wedge \neg p)$
- Tábló konstruálás:



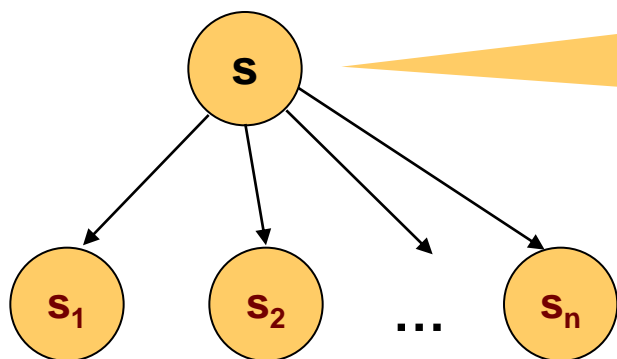
A tábló kiterjesztése PLTL-re

- Modellellenőrzés: Ellenpéldát keresünk adott kifejezéshez, tehát az eredeti PLTL kifejezés negáltjából készül a tábló!
 - A negált kifejezésből kell negált normál formát képezni
 - 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 (állapotokon)
 - 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, ld. szintaxis)

Felbontás az X operátor esetén



amennyiben a modellben:

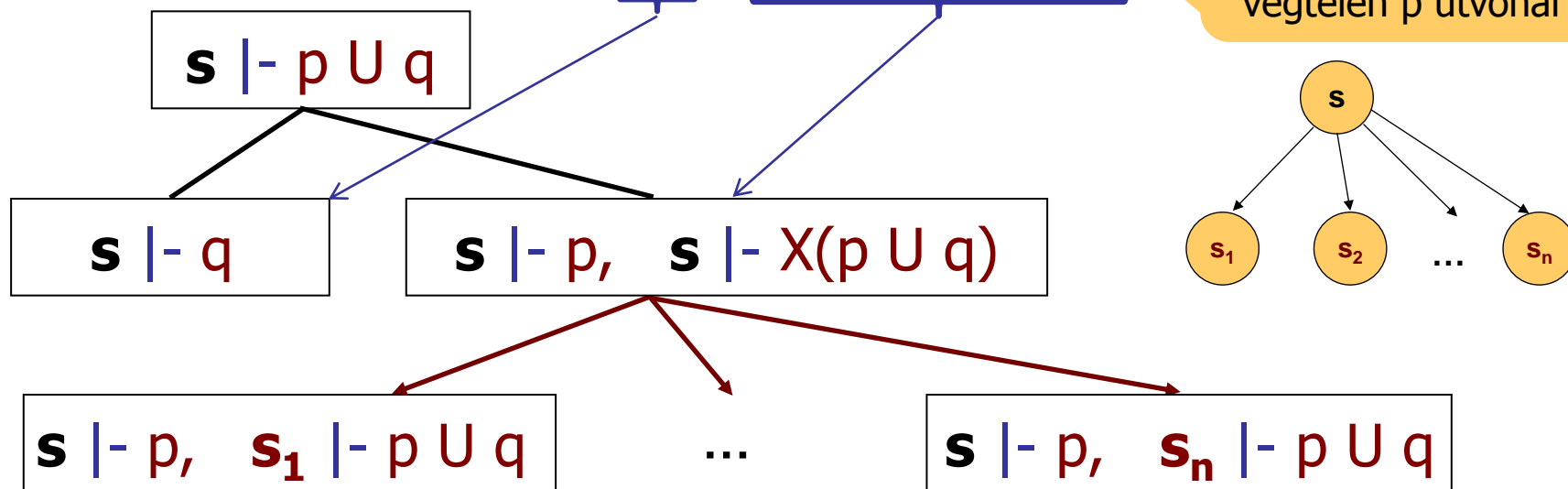


Közvetlen
ellentmondás,
ha s -nek nincs
rákövetkező
állapota!

Felbontás az U operátor esetén

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

Külön figyelni kell:
véges útvonal,
végtelen p útvonal



- A felbontás meddig folytatódik?

- **Ellentmondásra jutunk:**

- Atomi kijelentésre vonatkozó lokális állítás nem teljesül
- X operátor van, de az útvonal véget ér q teljesülése nélkül
- Ciklus alakul ki p teljesülésével, de q teljesülése nélkül

- **Sikeres ágak:**

- Atomi kijelentésekre vonatkozó állítások listája teljesül
- Ciklus alakul ki, és nincs ellentmondás

Egy speciális operátor: R

- Negált normál formára hozás az U operátor esetén:

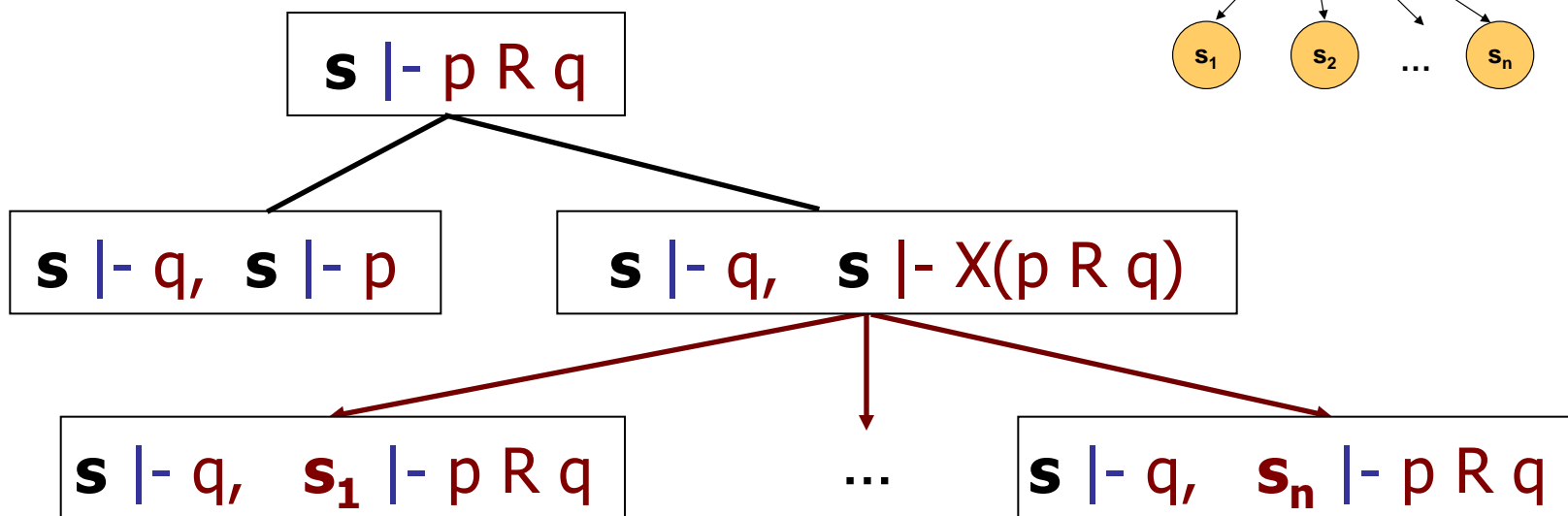
$$\neg(p \cup q) = ?$$

- Bevezethető az U operátor „duálisa”, az R (Release)

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

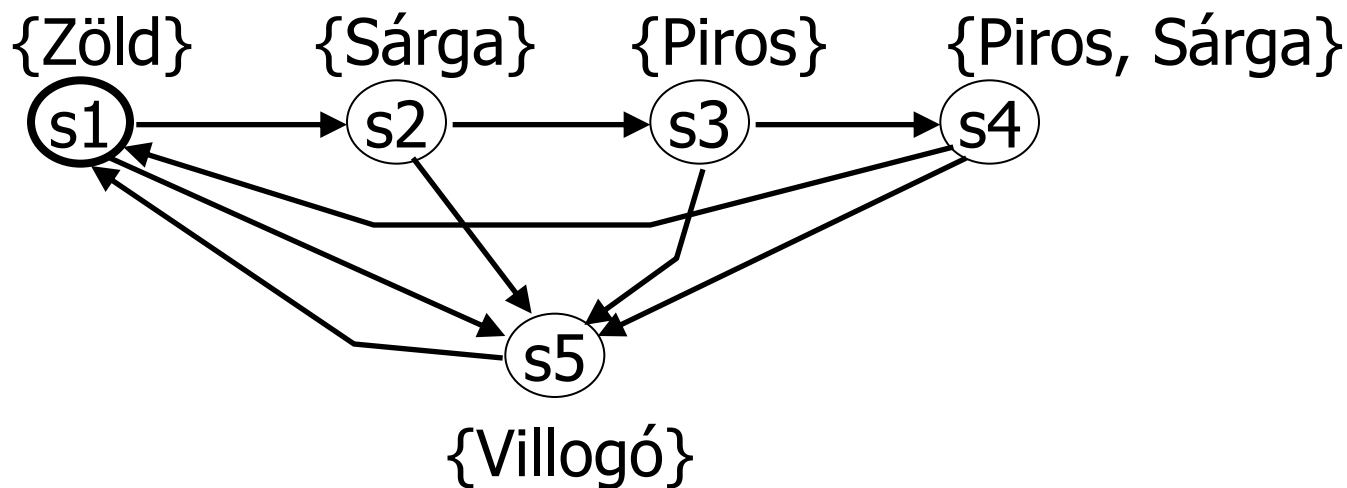
- Felírható: $p R q = q \wedge (p \vee X(p R q))$

- Az R operátor tablója:



Egy példa: A modell és a PLTL kifejezés

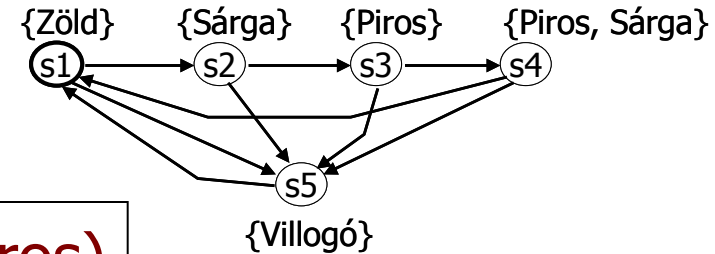
- A közlekedési lámpa vezérlő egy modellje (KS)
- Igaz-e, hogy ha a lámpa a kezdeti állapotban **Zöld**, akkor előbb-utóbb **Piros** lesz?
 - Az ellenőrizendő PLTL kifejezés: **Zöld** \Rightarrow F **Piros**



A modell alapján „kézzel” tudunk-e ellenpéldát adni?

Egy példa: A kifejezés tablója (1)

- A kifejezés negálása: $s1 \vdash \neg(\text{Zöld} \Rightarrow \text{F Piros})$
- Negált normál forma ($P \Rightarrow Q = \neg P \vee Q$ alapján):
 $\neg(\text{Zöld} \Rightarrow \text{F Piros}) = \text{Zöld} \wedge \neg \text{F Piros} = \text{Zöld} \wedge G(\neg \text{Piros})$
- A tábló konstruálása:



s1 állapotban van
Zöld címke

$s1 \vdash \text{Zöld} \wedge G(\neg \text{Piros})$

$s1 \vdash \text{Zöld}, s1 \vdash G(\neg \text{Piros})$

$s1 \vdash G(\neg \text{Piros})$

Egyszerűsített
jelölés
($s1 \vdash \text{Zöld}$
teljesült, kimarad)

Folytatódik a következő dián!

Egy példa: A kifejezés táblója (2)

s1 állapotban nincs Piros

$s1 \models G(\neg \text{Piros})$

$s1 \models \neg \text{Piros}, XG(\neg \text{Piros})$

s1 állapot után s2 vagy s5 jöhet

$s1 \models XG(\neg \text{Piros})$

s2 állapotban nincs Piros

$s2 \models G(\neg \text{Piros})$

$s2 \models \neg \text{Piros}, XG(\neg \text{Piros})$

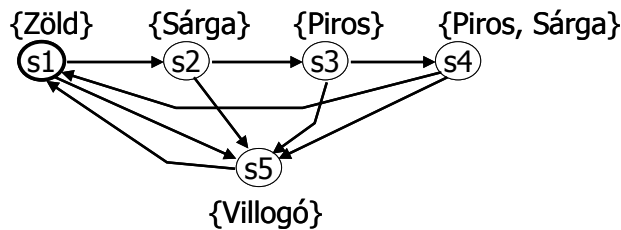
s2 után s3 vagy s5

$s2 \models XG(\neg \text{Piros})$

Ellentmondásos ág, s3-ban van Piros

$s3 \models G(\neg \text{Piros})$

$s3 \models \neg \text{Piros}, XG(\neg \text{Piros})$



$s5 \models G(\neg \text{Piros})$

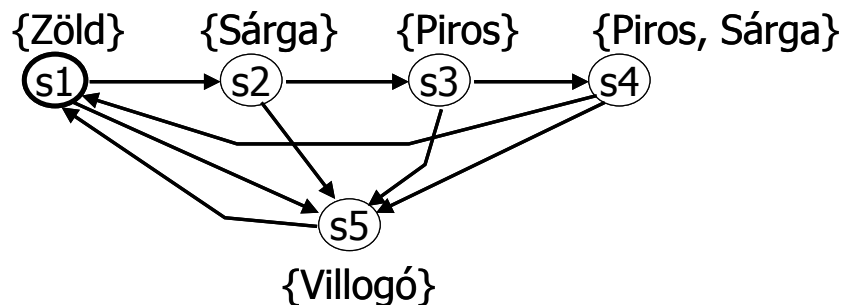
$s5 \models \neg \text{Piros}, XG(\neg \text{Piros})$

$s5 \models XG(\neg \text{Piros})$

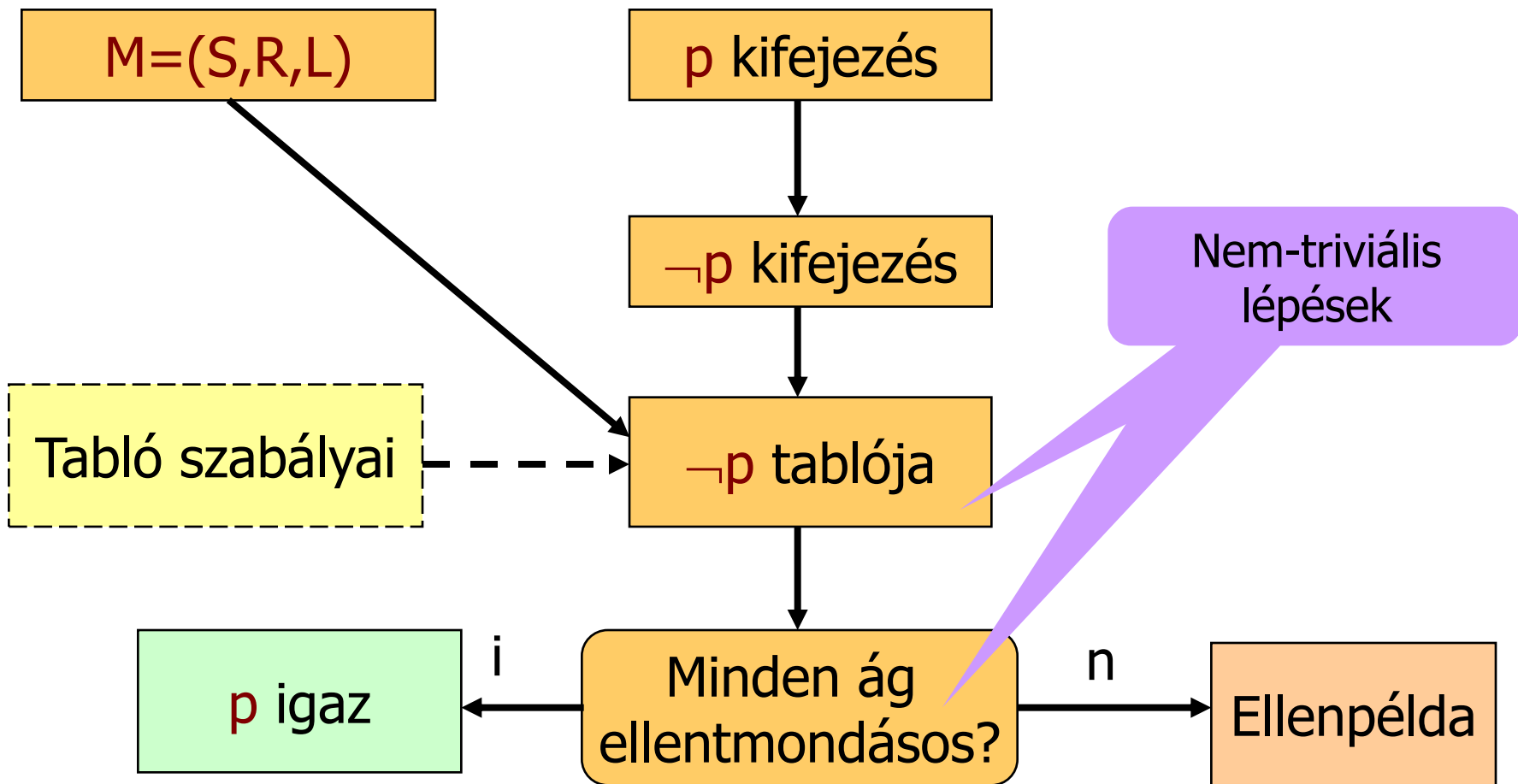
Ellentmondás nélküli ciklusok:
s1, s5, ...
s1, s2, s5, ...

Egy példa: A modellellenőrzés eredménye

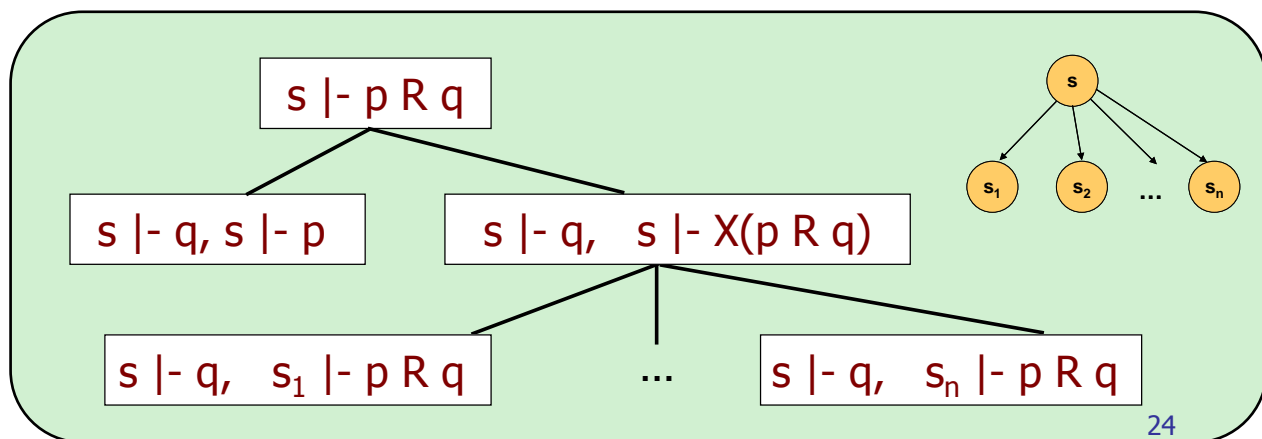
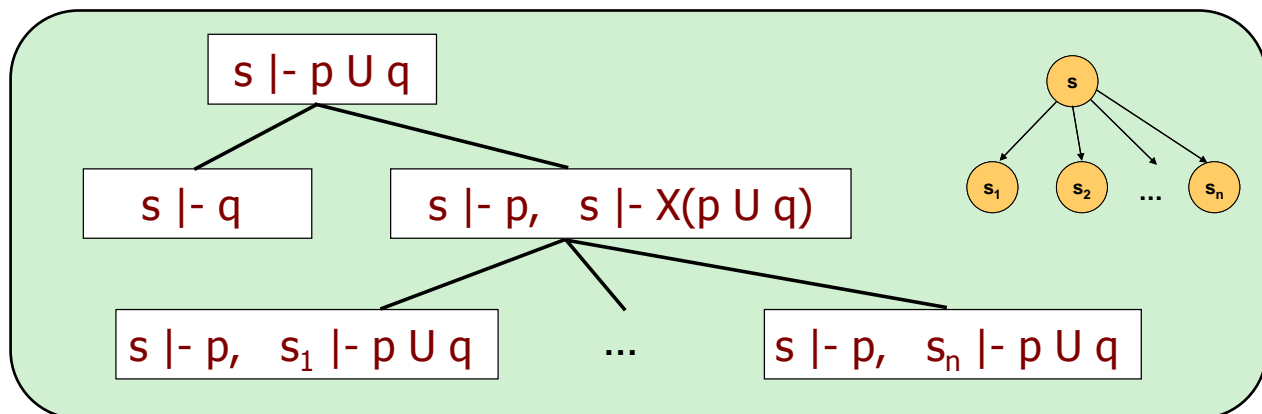
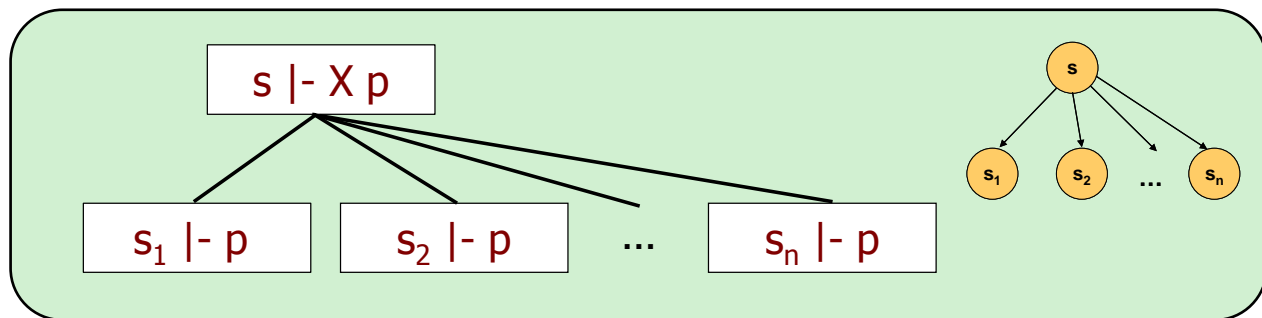
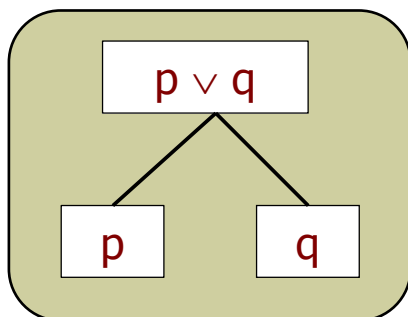
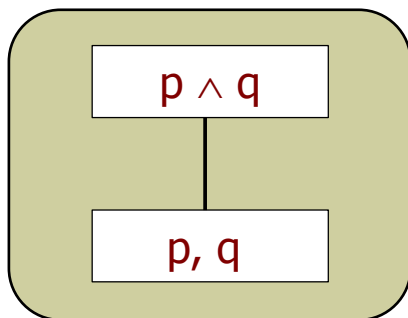
- A tábló eredménye a **negált kifejezésre**:
 - Egy ellentmondásos ág (a negált kifejezés nem teljesül)
 - Két ellentmondás nélküli ciklus (a negált kifejezés teljesül)
- Következtetés:
 - Vannak olyan lefutások, ahol a **negált kifejezés teljesül**:
Ciklus 1: s1, s2, s5, ...
Ciklus 2: s1, s5, ...
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



A tábló módszer (összefoglalás)



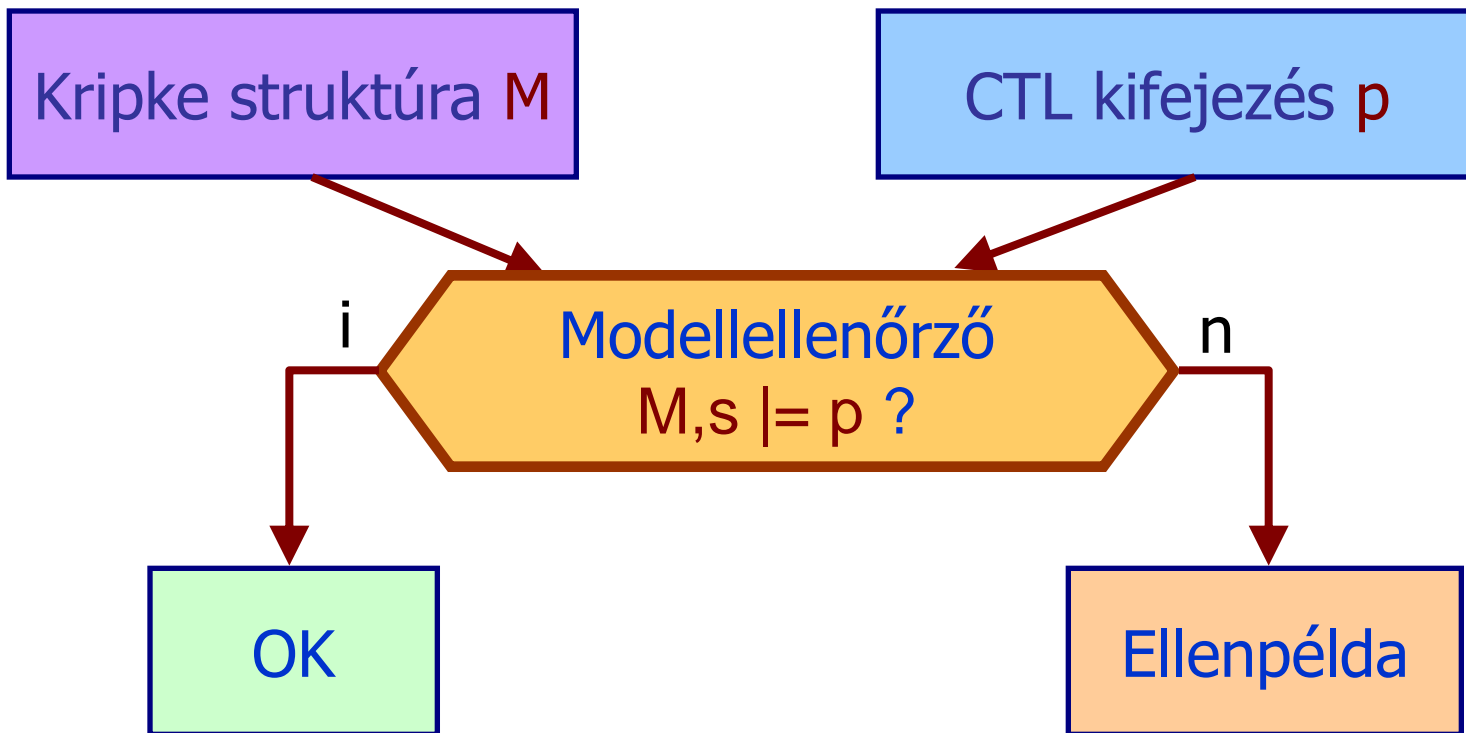
Tabló felbontási szabályok (összefoglalás)



A negált kifejezés felbontásaként kapott sikeres ágak az ellenpéldák

CTL modellellenőrzés szemantika alapon

A modellellenőrzés feladata



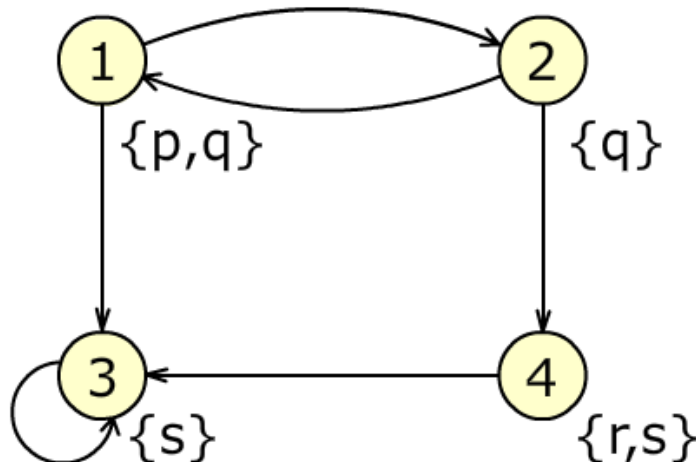
CTL temporális logika gyakorlása

Szegedi Tudományegyetem

<http://www.inf.u-szeged.hu/~zlnemeth/verifikacio/ctl.html>

- Interaktív mód, formula kiválasztása, állapotokra kattintás
- Helyes-e az eredmény: „Tesztelés”
- Mi a helyes eredmény: „Megoldás”

CTL formulák szemantikája



Bemutató mód Interaktív mód Leírás

Formula:
Formula kiválasztása ◀ ▶

Jelölje be a megfelelő állapotokat! A **tesztelés** gombbal ellenőrizheti megoldását. A **megoldás** gomb megmutatja a helyes megoldást.

Alapötlet: Állapotok címkézése

- **Globális modellellenőrzés p kifejezésre:**
 - Címkézzük fel p -vel a modell összes olyan állapotát, ahol a p kifejezés igaz
 - A modellen igaz p , ha a kezdőállapoton szerepel a p címke
- A címkézés az eredeti kifejezés rész-kifejezéseivel kezdve történik
 - Első lépés: Első rész-kifejezések az **atomi kijelentések**, ezek már szerepelnek címkeként
 - Következő lépések: **Összetett** rész-kifejezések p -ben, amik egy operátor használatával az **eddig már címkeként megjelent** rész-kifejezésekből adódnak
 - A címkézés vége: Az eredeti p kifejezés lesz a címke

CTL modellellenőrzés állapot címkézéssel

- Állapotok címkézése: ahol **igaz** egy adott (rész)kifejezés
- Összetett kifejezés esetén hogyan történik a címkézés?
 - Kifejezések felbontása azok szintaktika struktúrája alapján, „belülről kifelé” **haladva**, egy-egy újabb operátort alkalmazva:

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

Első lépés: P, Q, R

Következő: $E(Q \cup R)$

Következő: $P \wedge E(Q \cup R)$

Utolsó: $AF(P \wedge E(Q \cup R))$

- **Algoritmus az összetett kifejezés felbontása alapján:**
 - **Kiindulás:** KS címkézve van **atomi kijelentésekkel**
 - **Tovább lépés:** Címkézés az egyre összetettebb részkifejezésekkel
 - **Szabályok:** Ha p illetve q címkék már vannak, akkor megadható, hol lehet $\neg p$, $p \wedge q$, $EX p$, $AX p$, $E(p \cup q)$, $A(p \cup q)$ címke
 - Így haladunk egy összetett kifejezésben „belülről kifelé”

Szabályok: Atomi kijelentések és Boole operátorok

- P atomi kijelentés azokban az s állapotokban igaz, ahol $P \in L(s)$
 - A modellben P szerepel s címkéi között
- $\neg P$ azokban az s állapotokban igaz, ahol $P \notin L(s)$
 - Ahol $P \notin L(s)$, az állapot $\neg P$ kifejezéssel címkézhető
- $p \wedge q$ azokban az s állapotokban igaz, ahol p és q is igaz
 - Egy állapot címkézése lehet $p \wedge q$, ha címkéi között már szerepel p és q
- Temporális operátorok: EX , AX , $E(U)$, $A(U)$
 - Bonyolultabb szabályokat igényel a címkézés!

Szabályok: 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



Szabályok: 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 (ld. EX) már címkézett $E(p \cup q)$ -val
- Iteráció adódik, csak éppen az átmeneteken „visszafelé”:
 - 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

$E(p \cup q)$ modellellenőrzés alapötlete

$$\underline{E(p \cup q)} = \underline{q} \vee (\underline{p} \wedge \underline{EX E(p \cup q)})$$

Hová
rakható
 $E(p \cup q)$
címke?

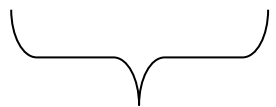
Ahol q
címke
már
van ...

vagy

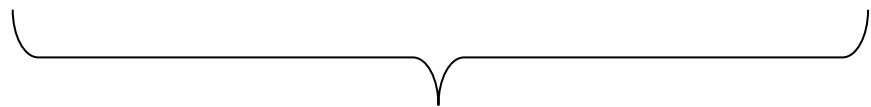
ahol p
címke
már
van ...

és

van legalább egy
rákövetkezője, ahol
már van $E(p \cup q)$
címke

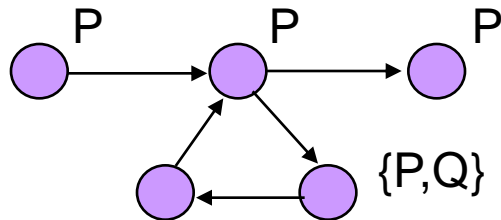


Címke
ráhelyezés
első lépése



Iteratívan bővíthető
a címkehalmaz
(amíg bővül)

Példa: Az $E(P \cup Q)$ címkézés iterációja

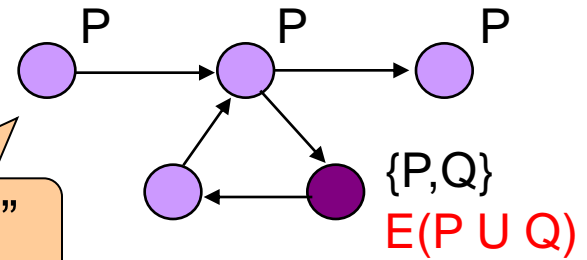


Kripke struktúra a kezdő címkézéssel

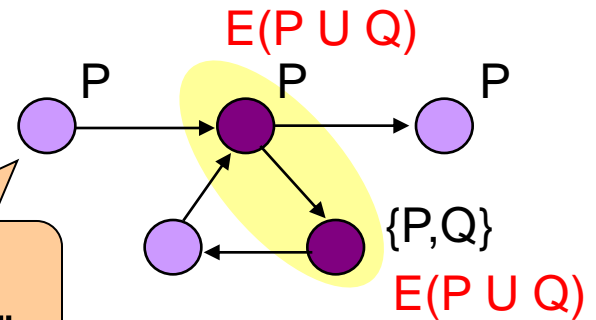
$$E(P \cup Q) = Q \vee (P \wedge EX E(P \cup Q))$$

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

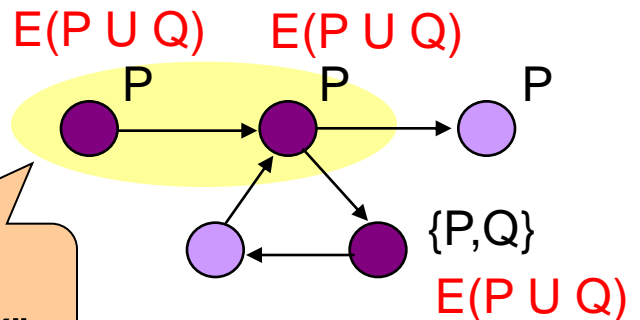
Első lépés: „Q”



Második lépés: „ $P \wedge EX$ ”



Harmadik lépés: „ $P \wedge EX$ ”



Szabályok: 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 (ld. AX) 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!

Iteráció halmazműveletekkel: Definíciók

- Ismétlés: Címkézési szabályok

- 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

$pre_E(\text{címkézett})$

- 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

$pre_A(\text{címkézett})$

- Ezek alapján a címkézett állapotokat megelőző megfelelő állapotok valamelyikére tehető az új címke

- Hogyan definiálhatók a megfelelő megelőző állapotok?

- Már címkézett Z állapotthalmaz alapján:

$pre_E(Z) = \{s \in S \mid \text{létezik olyan } s', \text{ hogy } (s, s') \in R \text{ és } s' \in Z\}$

$pre_A(Z) = \{s \in S \mid \text{minden } s'\text{-re, ahol } (s, s') \in R: s' \in Z\}$

Legalább egy
rákövetkezője
címkézett (Z -ben)

Minden
rákövetkezője
címkézett (Z -ben)

Iteráció halmazműveletekkel: Algoritmus

- A címkézés bővítése halmazműveletekkel történik
- Definiált jelölések:
 - $\text{pre}_E(Z)$: olyan állapotok, amelyek legalább egy rákövetkezője címkézett (Z-ben)
 - $\text{pre}_A(Z)$: olyan állapotok, amelyek minden rákövetkezője címkézett (Z-ben)
- $E(p \cup q)$ címkézési szabály: 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
- $E(P \cup Q)$ címkézési szabály halmazműveletekkel:
 - Kezdőhalmaz: $Z_0 = \{s \mid Q \in L(s)\}$, azaz Q -val címkézettek
 - Címkézés bővítése: $Z_{i+1} = Z_i \cup (\{s \mid P \in L(s)\} \cap \text{pre}_E(Z_i))$

Eddig
címkézettek uniója ...

... P-vel
címkézettek és ...

... legalább egy rákövetkezője
már címkézett

- Iteráció vége: Ha $Z_{i+1} = Z_i$, azaz már nem bővül a halmaz

CTL modellellenőrzés: Összefoglalás

- Globális modellellenőrzés:
 - Állapotok címkézése azokkal a (rész)kifejezésekkel, amelyek igazak az adott állapotban
 - Címkézés egyre összetettebb kifejezésekkel („belülről kifelé”), az atomi kijelentésekből indítva az összetettebb kifejezések felé
- Címkézés egy részkifejezéssel:
 - Az előző lépésben adott címkézés felhasználása az operátorok szemantikája alapján képzett szabályok szerint
 - EX, AX esetén: Megelőző állapot vizsgálata és címkézése
 - 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 állapotokat címkézve
 - Iteráció vége: Nem nő a címkézett állapotok halmaza

A bevezető példa kifejtése

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

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

Q és R címkék
a KS-ban

Inkrementális címkézés: $E(. \cup .)$
Az iteráció végén megjelenik
az $E(Q \cup R)$ címke

Itt P -vel és $E(Q \cup R)$ -val címkézett
állapothalmazok metszete,
építve a már meglévő $E(Q \cup R)$
címkékre:
Megjelenik a $P \wedge E(Q \cup R)$ címke

Inkrementális címkézés: AF alapján,
építve a már meglévő $P \wedge E(Q \cup R)$
címkékre:
Megjelenik az $AF(P \wedge E(Q \cup R))$ címke.
Ez a kezdőállapotra ellenőrizhető.

Gyakorló feladat

- Egy speciális jelzőlámpa 3 égőt tartalmaz: piros, sárga és zöld.
 - A jelzőlámpa alaphelyzetében mindhárom égő kikapcsolt.
 - Bekapcsolás után rögtön a piros égő világít.
 - Innen két választási lehetőség adódik a továbblépésre: egyik esetben a lámpa piros-sárgára (mindkettő világít), másik esetben zöldre vált.
 - A választástól függően a piros-sárga után következik a zöld, míg a zöld után újra a piros, majd ezekből az ismert állapotokból folytatódik tovább a működés.
- A jelzőlámpa alaphelyzetéből kiindulva teljesül-e:
$$E((\neg \text{piros}) \cup (EX \text{zöld}))$$

Összefoglalás

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

Hogyan tehető hatékonyá ez az algoritmus?

PLTL modellellenőrzés: Automata alapú megközelítés

(Kitekintés, kiegészítő anyag)

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

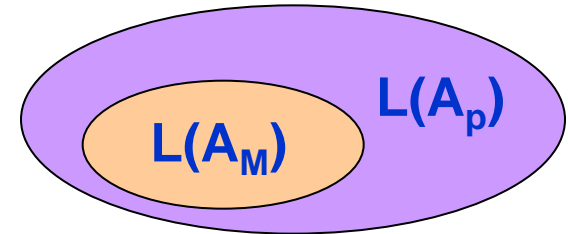
- Alkalmazás: Folyamatosan működő rendszerek
 - Nem ellenőrizhető, hogy a végállapot elfogadó-e
- Büchi elfogadási kritérium:
 - Egy beérkező $w=(a_0, a_1, a_2, \dots)$ betűsorozat hatására egy $r=(s_0, s_1, s_2, \dots)$ állapotsorozat
 - $\lim(r) = \{s \mid s \text{ előfordul végtelenül sokszor, azaz nincs olyan } j, \text{ hogy } \forall k > j: s \neq s_k\}$
 - Elfogadó a futás, ha $\lim(r) \cap F \neq \emptyset$
 - 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

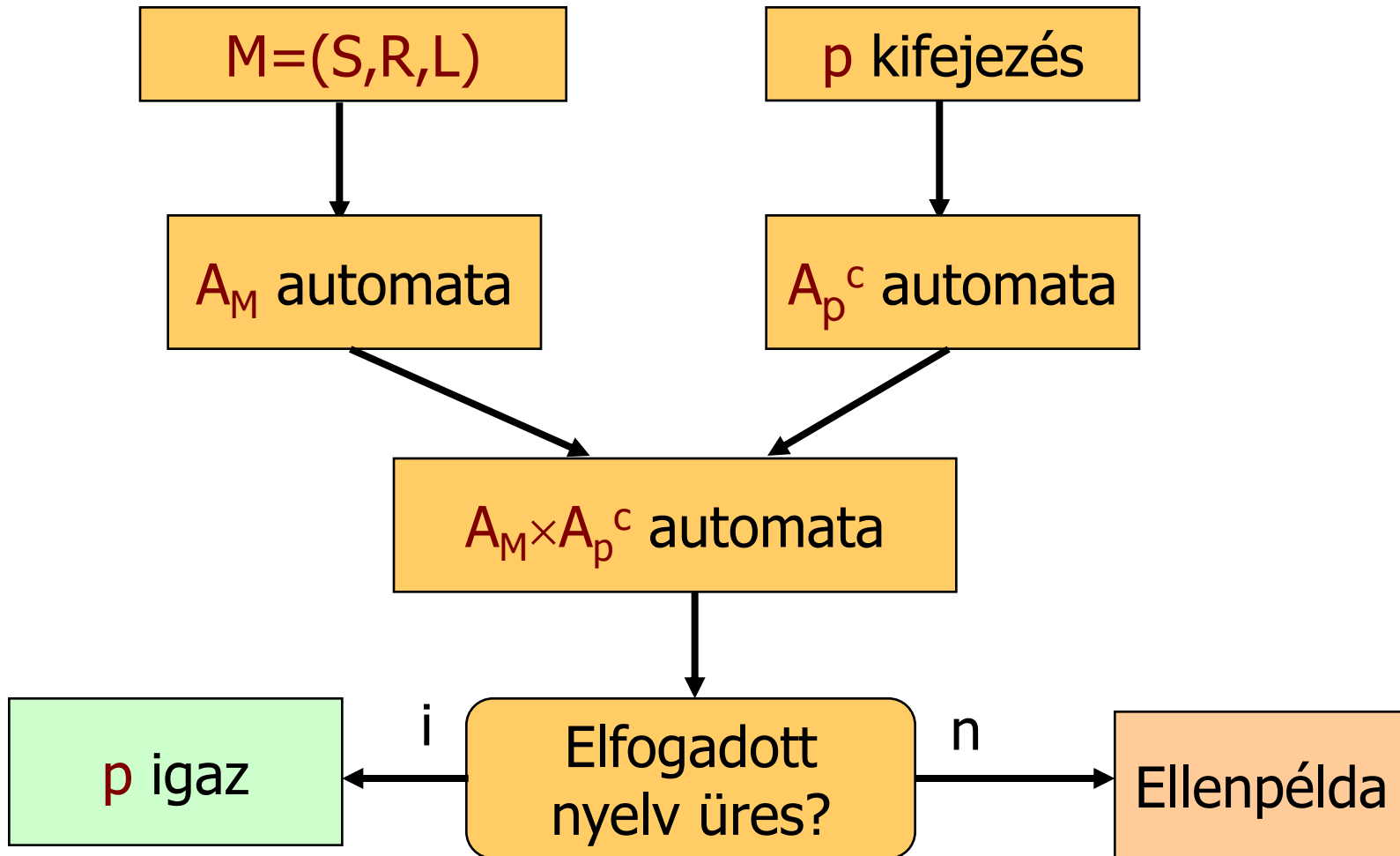
- A **KS** egy **s** állapothoz: $L(s)$ a betű a 2^{AP} ábécéből
 - Pl. {Piros, Sárga} az ábécé egy betűje
- A $\pi = (s_0, s_1, s_2, \dots, s_n)$ útvonal egy szót azonosít:
 $(L(s_0), L(s_1), L(s_2), \dots, L(s_n))$
- Két automatát kell konstruálni:
 - $M = (S, R, L)$ alapján egy A_M automata konstruálható, 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: mi kell teljesüljön egy adott állapotban, és mi a rákövetkezőben (ld. X után)

Modellellenőrzés az automatákkal

- Modellellenőrzési feladat: $L(A_M) \subseteq L(A_p)$, 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