Követelmények formalizálása: Elágazó idejű temporális logikák

dr. Majzik István
BME Méréstechnika és Információs Rendszerek Tanszék

Ismétlés: Mit szeretnénk elérni?

Alacsony szintű modellek:
- KS, LTS, KTS
- XTA időzített automata

Állapot elérhetőségi követelmények:
- Temporális logikák: lineáris / elágazó idejű

Rendszer modellje

Követelmény megadása

Automatikus modellellenőrző

i

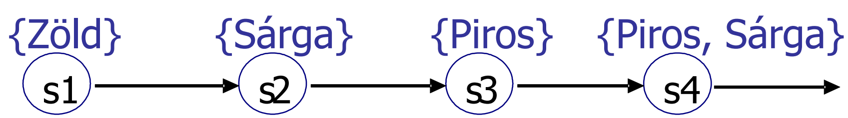
n

OK

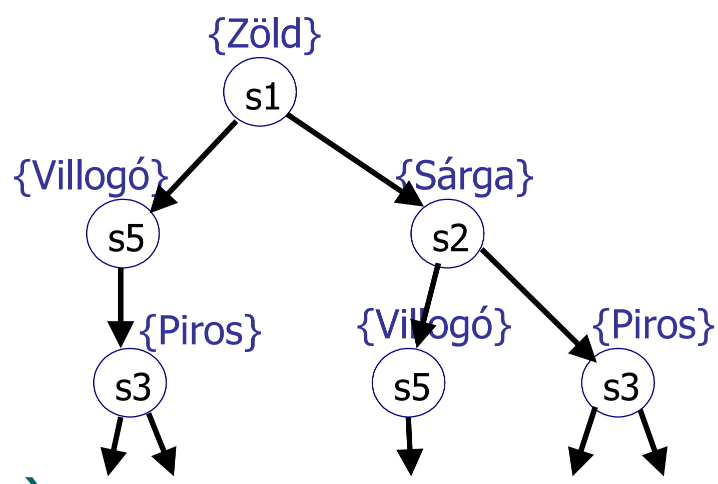
Ellenpélda

Ismétlés: Temporális logikák osztályozása

- Lineáris idejű:
  - A modell egy-egy végrehajtását (lefutását) tekintjük
  - Minden állapotnak egy rákövetkezője van
  - Logikai idő egy idővonal mentén (állapotsorozat)

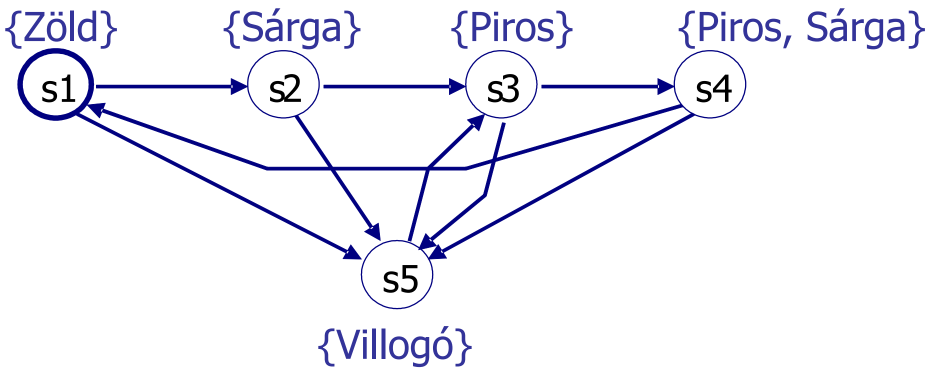


- Elágazó idejű:
  - A modell minden lehetséges végrehajtását tekintjük
  - Az állapotoknak több rákövetkezője lehet
  - Logikai idő elágazó idővonalak mentén jelenik meg (számítási fa)

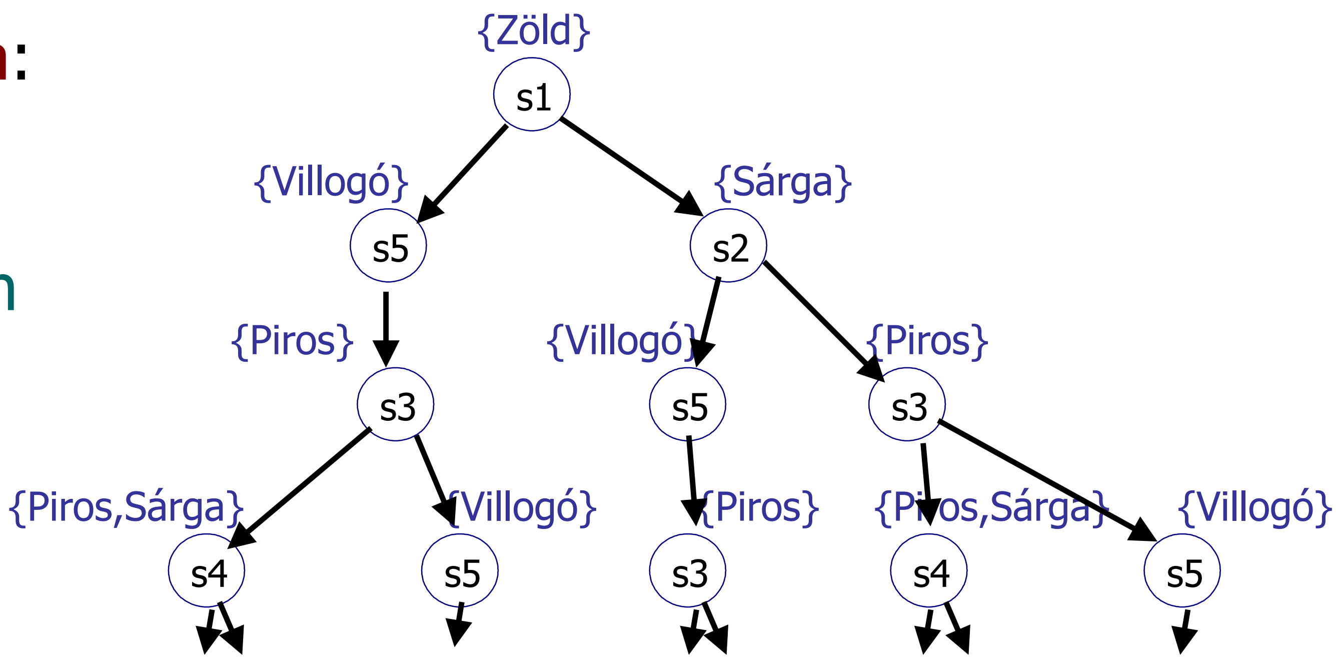


Számítási fa konstrukciója

Kripke-struktúra:



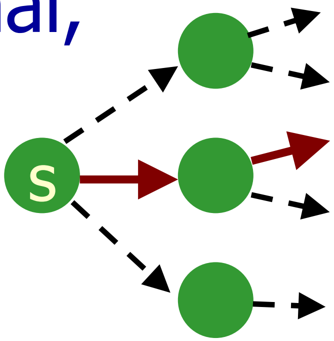
Számítási fa:
Lehetséges elágazások a modellben



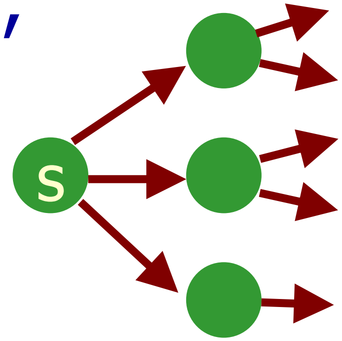
Elágazások vizsgálata

Egy-egy állapotban előírható,
hogy az útvonalakra vonatkozó p követelmény
hogyan teljesüljön az onnan kiinduló útvonalakra:

- E p (Exists p): Létezzen legalább egy útvonal, ahol a p követelmény teljesül
  - Egy útvonalat ír elő (lehet több is)
  - Egzisztenciális útvonal kvantor



- A p (forAll p): Minden útvonalra álljon fenn, hogy a p követelmény teljesül
  - Minden útvonalra előírást tesz
  - Univerzális útvonal kvantor



Elágazó idejű temporális logikák

- CTL*: Computational Tree Logic *
  Útvonal kvantorok (E, A) és
  útvonalakon értelmezett temporális operátorok (X, F, G, U)
  tetszőleges értelmes kombinációja

- CTL: Computational Tree Logic
  Útvonalakon értelmezett temporális operátorokat (X, F, G, U)
  közvetlenül meg kell előznie útvonal kvantoroknak (E, A)
  - Útvonalakon értelmezett temporális operátorok egymással nem kombinálhatók
  - „Összenőnek” az útvonal kvantorok a temporális operátorokkal (pl. EX, AX, EF, AF, …)

CTL*: Computational Tree Logic *

CTL* operátorok (informálisan)

- Útvonalak kvantorai (állapotokon értelmezett):
  - A: „for All paths”,
    minden lehetséges útra az adott állapotból kiindulva
  - E: „Exists a path”, „for some path”,
    legalább egy útra az adott állapotból kiindulva
- Útvonalakon értelmezett operátorok:
  - X p: „neXt”, a következő állapotban p igaz
  - F p: „Future”, valamikor az útvonal egy állapotán p igaz
  - G p: „Globally”, az útvonal minden állapotán p igaz
  - p U q: „p Until q”, az útvonal egy állapotán igaz lesz q, és addig minden állapotban igaz p

CTL* kifejezések

$A(p \Rightarrow F\ q)$ ld. például $A(Start \Rightarrow F\ End)$

Minden útvonalra igaz, hogy …

amennyiben p fennáll az útvonal elejétől, …

akkor ezt előbb-utóbb olyan állapot fogja követni …

ahonnan nézve (a további viselkedésre) q fennáll.

Példa CTL* kifejezések

- $E(p \wedge G\ q)$

  Igaz, ha létezik olyan útvonal, hogy
  ennek kezdőállapotán p fennáll
  és az útvonal minden állapotán q is fennáll

- $A(XXX\ p \vee F\ q)$

  Igaz, ha minden útvonalra teljesül, hogy
  – ennek negyedik állapotán fennáll p,
  – vagy valamikor az útvonalon fennáll q

A CTL* formális szintaxisa és szemantikája

CTL* szintaxis

- Állapot-kifejezések: Állapotokon kiértékelhető
  - **S1**: Minden P atomi kijelentés egy állapot-kifejezés
  - **S2**: Ha p és q állapot-kifejezések, $\neg$p és p$\wedge$q is
  - **S3**: Ha p útvonal-kifejezés, akkor E p és A p állapot-kifejezések.
- Útvonal-kifejezések: Útvonalakon kiértékelhető
  - **P1**: Minden állapot-kifejezés útvonal-kifejezés
  - **P2**: Ha p és q útvonal-kifejezések, akkor $\neg$p és p$\wedge$q is
  - **P3**: Ha p és q útvonal-kifejezések, akkor X p és p U q is

Érvényes CTL* kifejezések:
A szabályok alapján generált állapot-kifejezések

CTL* szemantika: Jelölések

- $M = (S, R, L)$ Kripke-struktúra
- $\pi = (s_0, s_1, s_2, \ldots)$ az M egy útvonala, ahol s_0 a kezdőállapot és $\forall i \geq 0$: $(s_i, s_{i+1}) \in R$

  valamint $\pi^i = (s_i, s_{i+1}, s_{i+2}, \ldots)$ a π útvonal szuffixe i-től

- $M,\pi \models p$ jelöli (ahol p útvonal-kifejezés): az M modellben a π útvonalon igaz p
- $M,s \models p$ jelöli (ahol p állapot-kifejezés): az M modellben az s állapotban igaz p

CTL* szemantika: Állapot-kifejezések

- **S1**:

  M,s |= P a.cs.a. $P \in L(s)$

- **S2**:

  M,s |= $\neg$p a.cs.a. M,s |= p nem igaz

  M,s |= p$\wedge$q a.cs.a. M,s |= p és M,s |= q

- **S3**:

  M,s |= E p (ahol p útvonal-kifejezés)

  a.cs.a. létezik $\pi=(s_0, s_1, s_2,\ldots)$ útvonal M-ben

  $s= s_0$ mellett, hogy M,π |= p.

  M,s |= A p (ahol p útvonal-kifejezés)

  a.cs.a. minden $\pi=(s_0, s_1, s_2,\ldots)$ útvonalra M-ben

  ahol $s= s_0$ fennáll igaz, hogy M,π |= p.

CTL* szemantika: Útvonal-kifejezések

- **P1**:

  $M,\pi \models p$ (p állapot-kifejezés) a.cs.a. $M, s_0 \models p$

- **P2**:

  $M,\pi \models \neg p$ a.cs.a. $M,\pi \models p$ nem igaz

  $M,\pi \models p \wedge q$ a.cs.a. $M,\pi \models p$ és $M,\pi \models q$

- **P3**:

  $M, \pi \models X\, p$ a.cs.a. $M,\pi^1 \models p$

  $M, \pi \models p\, U\, q$ a.cs.a

  $$\exists j \geq 0 : (M, \pi^j \models q \text{ valamint } \forall 0 \leq k < j:\ M, \pi^k \models p)$$

Háttér: Az ellenőrzés komplexitása

- Worst-case időkomplexitás: Legalább $O(|S|^2 \times 2^{|p|})$
  - $|S|^2$ az átmenetek száma a modellben (Kripke-struktúrában) worst case esetben
  - $|p|$ az operátorok száma a temporális logikai kifejezésben
- Az exponenciális komplexitás riasztó
  - Bár a temporális logikai kifejezések tipikusan rövidek
- Célkitűzés: CTL* egyszerűsítése
  - Gyakorlati problémák esetén használható maradjon
  - Az ellenőrzés worst-case időkomplexitása csökkenjen

CTL: Computational Tree Logic

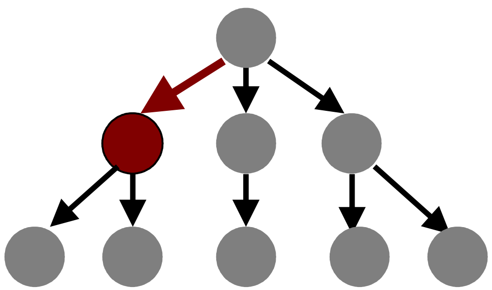
CTL operátorok (informális bevezető)

Állapotokon értelmezhető összetett operátorok:

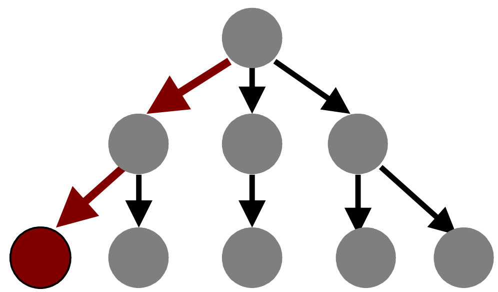
- EX p: létezik útvonal, aminek következő állapotán p
- EF p: létezik útvonal, aminek jövőbeli állapotán p
- EG p: létezik útvonal, aminek minden állapotán p
- E(p U q): létezik útvonal, amin p amíg q

- AX p: minden útvonal következő állapotán p
- AF p: minden útvonal egy-egy jövőbeli állapotán p
- AG p: minden útvonal minden állapotán p
- A(p U q): minden útvonalon p amíg q

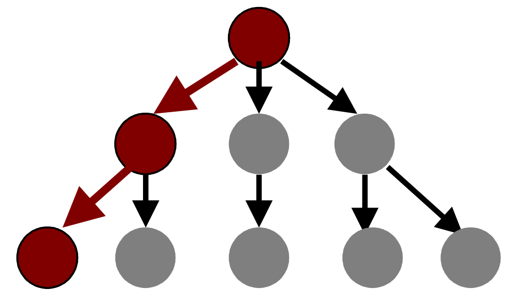
CTL operátorok (példák)



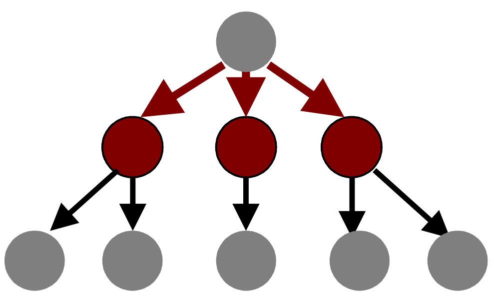
EX P



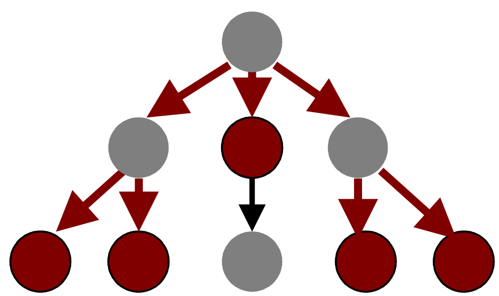
EF P



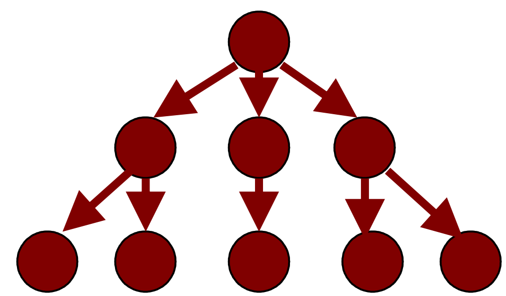
EG P



AX P



AF P



AG P

CTL kifejezések (példák)

- AG EF p

  Minden útvonal minden állapotára: onnan nézve van olyan útvonal, ahol előbb-utóbb p igaz

  - Példa: AG EF Reset (mindenhonnan elérhető a Reset állapot)

- AG AF p

  Bárhonnan indulva: mindenképpen elérhető olyan állapot, ahol p igaz

  - Példa: AG AF Terminated (mindenhonnan Terminated lesz)

- AG (p $\Rightarrow$ AF q)

  Bárhonnan indulva teljesül, hogy ha ott p igaz, akkor valamikor mindenképpen elérünk olyan állapotba, ahol q igaz.

  - Példa: AG (Request $\Rightarrow$ AF Reply)

CTL kifejezések (példák)

- EF AG p

  Lehetséges, hogy a rendszer olyan állapotba kerül, hogy utána p minden állapotban igaz lesz

- AF AG p

  Bármely úton haladva eljutunk olyan állapotba, hogy utána p mindig igaz lesz

  - Példa: AF AG Normal

- AG (p $\Rightarrow$ A (p U q))

  Bármelyik elérhető állapotban ha p igaz, akkor minden úton p fennáll q eléréséig

  - „p fennáll q eléréséig” pontosabban: elérünk egy olyan állapotba, ahol q igaz, és addig minden állapotban p igaz

Követelmények formalizálása: Egy példa

- Két processzből álló rendszer: P1 és P2
- Processz állapotok a követelmények szempontjából:
  - Kritikus szakaszban van: C1, C2
  - Nem-kritikus szakaszban van: N1, N2
  - Kritikus szakaszba belépésre kész: W1, W2
- Atomi kijelentések:
  AP = {C1, C2, N1, N2, W1, W2}

Követelmények formalizálása: Egy példa (folytatás)

- Egyszerre csak egy processz lehet a kritikus szakaszban:
  $AG\ (\neg(C1 \wedge C2))$
- Ha egy processz be akar lépni a kritikus szakaszba, akkor előbb-utóbb mindig beléphet:
  $AG\ (W1 \Rightarrow AF(C1))$
  $AG\ (W2 \Rightarrow AF(C2))$
- A processzek mindig felváltva kerülnek a kritikus szakaszba; egyikük kilép majd a másik lép be:
  $AG(C1 \Rightarrow A(C1\ U\ (\neg C1 \wedge A((\neg C1)\ U\ C2))))$
  $AG(C2 \Rightarrow A(C2\ U\ (\neg C2 \wedge A((\neg C2)\ U\ C1))))$

Követelmények formalizálása: Egy példa (folytatás)

- Egyszerre csak egy processz lehet a kritikus szakaszban:
  AG $(\neg($C1 ∧

P1 nincs a kritikus szakaszban

egészen addig, amíg

- Ha egy pro… ritikus szakaszba, akkor előbb…

P2 lép a kritikus szakaszba

P1 van a kritikus szakaszban

(C1)…

(C2))

- A process…k mindig felváltva kerülne… a kritikus szakaszba; …gyikük k…lép majd a mási… lép be:

$$AG(C1 \Rightarrow A(C1\ U\ (\neg C1 \wedge A((\neg C1)\ U\ C2))))$$
$$AG(C2 \Rightarrow A(C2\ U\ (\neg C2 \wedge A((\neg C2)\ U\ C1))))$$

A CTL formális szintaxisa és szemantikája

CTL formális szintaxis (összefoglalva)

Állapot-kifejezések (CTL*-hoz képest változatlan):

- **S1**: Minden P atomi kijelentés egy állapot-kifejezés
- **S2**: Ha p és q állapot-kifejezések, $\neg$p és p$\wedge$q is
- **S3**: Ha p útvonal-kifejezés, akkor E p és A p állapot-kifejezések.

Útvonal-kifejezések (csak egy szabály):

- **P0**: Ha p és q állapot-kifejezések, akkor X p és p U q útvonal-kifejezések.

- Útvonal-kifejezések nem kombinálhatók
- Útvonal-kifejezéseket csak az **S3** szabály használja, így az X p és p U q útvonal-kifejezések elé csak valamelyik útvonal kvantor kerülhet („összenőnek”)

CTL formális szemantika

- Állapot-kifejezések:
  - **S1**, **S2**, **S3** szabályok (lásd CTL*) változatlanok

- Útvonal-kifejezések:

P0:

– M,π |= X p ahol p állapot-kifejezés
a.cs.a. M,s_1 |= p

– M,π |= p U q ahol p,q állapot-kifejezés a.cs.a.

$$\exists j \geq 0 : (M, s_j \models q \text{ valamint } \forall 0 \leq \mathrm{k} < \mathrm{j}\text{: } \mathrm{M}, \mathrm{s}_\mathrm{k} \models p)$$

Itt állapot-kifejezések vannak a szintaxis szabály szerint!

CTL és CTL* kifejezések

- „Kimaradt” CTL operátorok
  - EF p jelentése E (true U p)
  - AF p jelentése A (true U p)
  - EG p jelentése $\neg$AF ($\neg$p)
  - AG p jelentése $\neg$EF ($\neg$p)

- CTL* de nem CTL kifejezés
  - E(X Piros $\vee$ F Sárga)

    Boole operátor van útvonal-kifejezések között
  - A(X G (Piros $\wedge$ Sárga)),
    E(XXX Piros)

    Egymásba ágyazott útvonal-kifejezések vannak

Háttér: Az ellenőrzés komplexitása

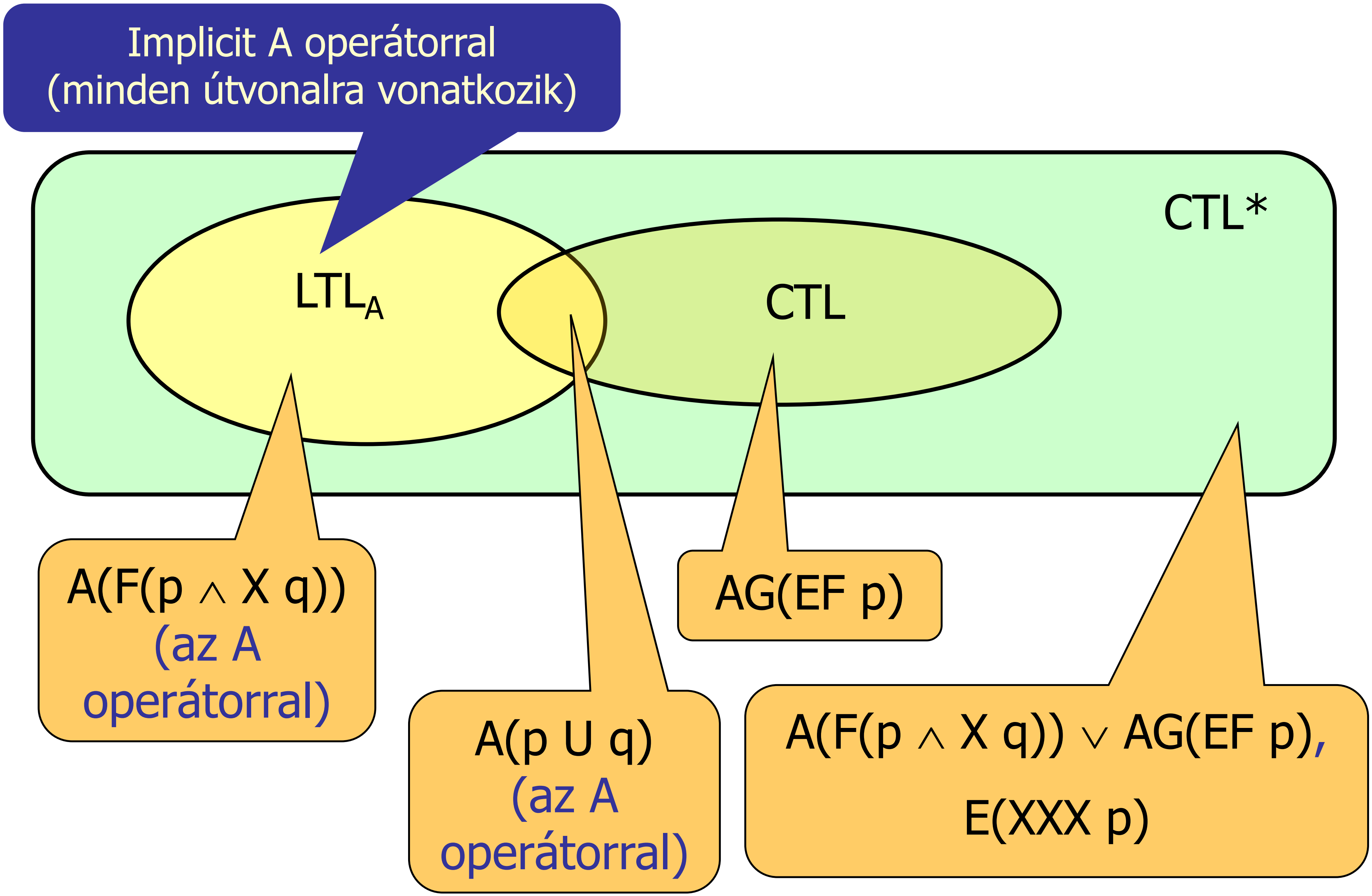
- CTL* worst-case időkomplexitás: $O(|S|^2 \times 2^{|p|})$
- CTL worst case időkomplexitás: $O(|S|^2 \times |p|)$
  - $|S|^2$ az átmenetek száma a modellben (Kripke-struktúra) worst case esetben
  - $|p|$ az operátorok száma a temporális logikai kifejezésben
- CTL*-nál kedvezőbb a CTL esetén:
  - Itt nincs $2^{|p|}$ tag
  - Sok gyakorlati követelmény esetén jól használható
    - Biztonsági követelmények: AG
    - Élőségi követelmények: EF, AF
  - De vannak kifejezőképességbeli korlátok

Temporális logikák kifejezőképessége

Kifejezőképesség

- Egy temporális logika kifejezőképessége nagyobb egy másikénál, ha
  - Minden olyan tulajdonságot formalizálni tud, amit a másik, valamint
  - képes olyan tulajdonságot is formalizálni, amit a másik nem
- Eddigi tapasztalatok:
  - Lineáris idejű logika nem tudja figyelembe venni a lehetséges elágazásokat
    (implicit „minden útra” jellegű vizsgálat lehetséges)
  - CTL kötöttebb, mint a CTL*,
    ezért kevesebb tulajdonság megfogalmazására képes
  - CTL* magába foglalja a lehetséges LTL kifejezéseket

LTL, CTL, CTL* kifejezőképessége



Kifejezőképesség - formálisan

- TL2 kifejezőképessége nagyobb vagy egyenlő (nem kisebb) mint TL1 kifejezőképessége, ha minden M modellre és annak minden s állapotára:

$$\forall p \in TL1:$$

$$\exists q \in TL2: \ (M, s \models p \ \Leftrightarrow \ M, s \models q)$$

- Ha ez kölcsönösen fennáll, akkor TL2 és TL1 azonos kifejezőképességűek.

Összefoglalás

- Elágazó idejű temporális logikák
  - CTL*
  - CTL (kötöttebb, de egyszerűbben ellenőrizhető)
- Formális szintaxis és szemantika
  - Precíz értelmezéshez szükséges
- Kifejezőképesség
- Modellellenőrzési feladat
  - Megoldás algoritmusa: Következő előadás!

Kitekintés

Sztochasztikus logikák:

- Megbízhatósági illetve időbeli követelményekhez
  - Pl.: Ha Hiba az aktuális állapot, akkor ez kevesebb, mint 30% valószínűséggel áll fenn 2 időegység múlva is
- CTL kiterjesztése: Continuous Stochastic Logic
  - Folytonos idejű Markov-láncokon értelmezett (nem Kripke-struktúra)
  - Valószínűségi kritériumok állapotok eléréséhez (állandósult állapot), útvonalak bejárásához
  - Idő kritériumok (időintervallumok) X és U operátorokhoz

Valósidejű logikák:

- Valósidejű rendszerek követelményeinek megadásához
  - Időintervallumok kezelése
  - Határidők kezelése