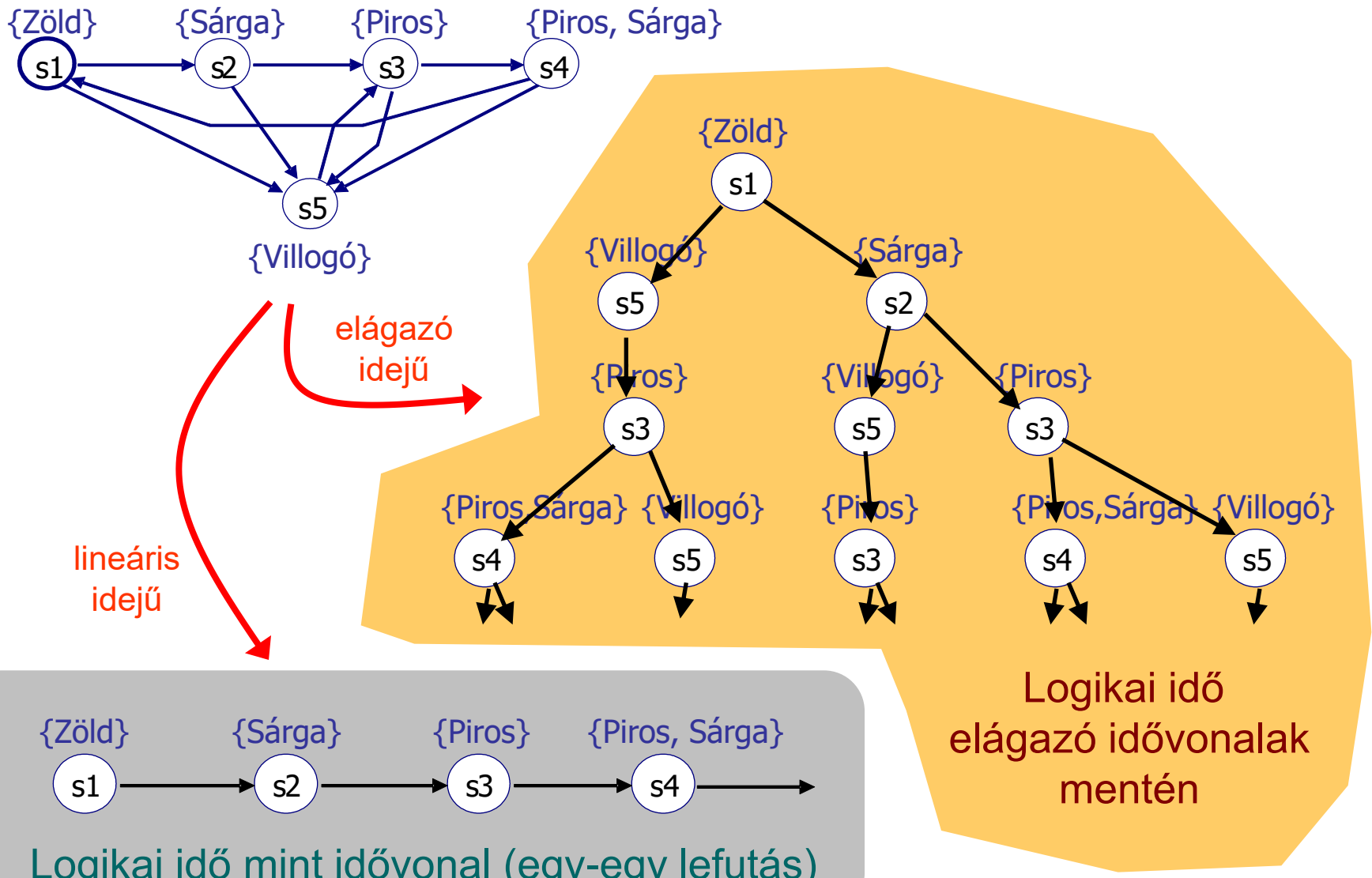
Elágazó idejű temporális logikák: Computational Tree Logic (CTL, CTL*)

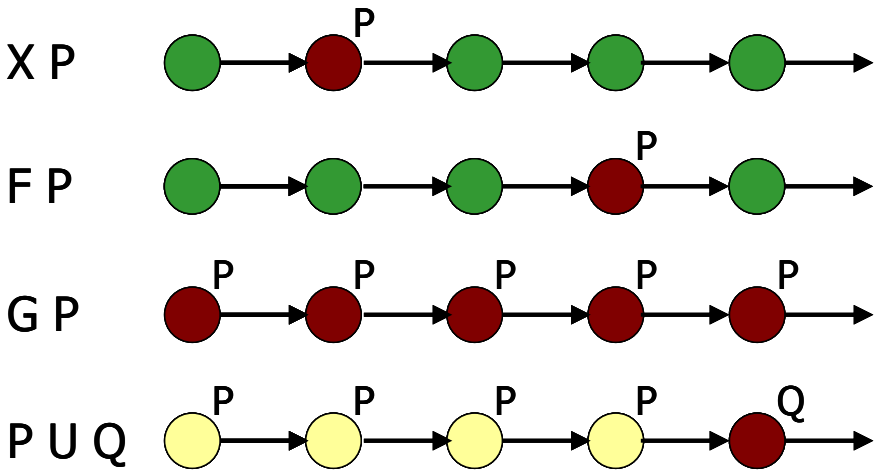
Majzik István
BME Méréstechnika és Információs Rendszerek Tanszék

Lineáris és elágazó idejű temporális logikák



Útvonalakra vonatkozó operátorok (lineáris TL)

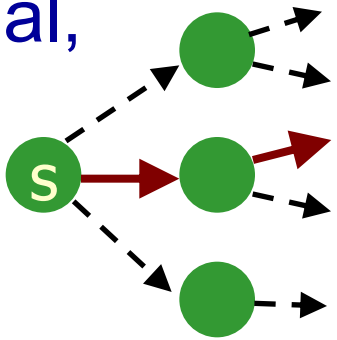
- Egy-egy útvonal állapotain értelmezett operátorok
- F, G, X, U informálisan:
  - F p: „Valamikor p”, egy elérhető állapotban igaz lesz p
  - G p: „Mindig p”, minden elérhető állapotban igaz lesz p
  - X p: „Következő p”, a következő állapotban igaz lesz p
  - p U q: „p amíg q”, egy elérhető állapotban igaz lesz q, és addig minden állapotban igaz p



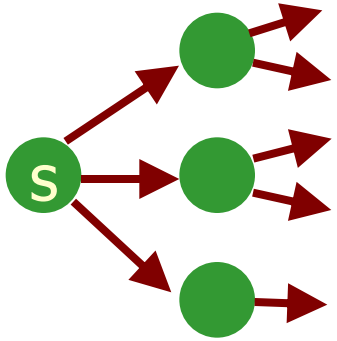
Elágazásokra vonatkozó operátorok (elágazó TL)

Egy-egy állapotban útvonal kvantorok:

- E p (Exists p): Létezik legalább egy útvonal, ahol a p követelmény teljesül
  - Egy lehetséges elágazás mentén vizsgál
  - Egzisztenciális operátor



- A p (forAll p): Minden útvonalra fennáll, hogy a p követelmény teljesül
  - Minden lehetséges elágazást magába foglal
  - Univerzális operátor



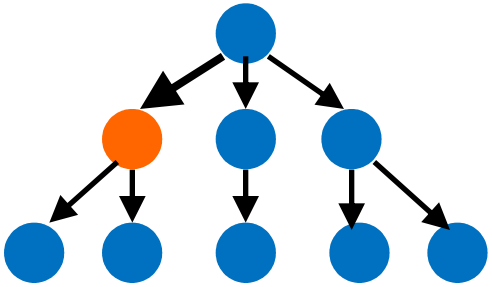
Elágazó idejű temporális logikák

- CTL*: Computational Tree Logic *

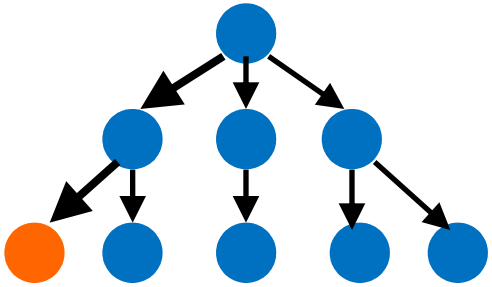
  Tetszőleges kombinációja a következőknek:
  - Útvonal kvantorok (E, A),
  - Útvonalon értelmezett (sorrendi) operátorok (F, G, X, U)

- CTL: Computational Tree Logic
  - Útvonalon értelmezett operátort mindig közvetlenül meg kell előznie útvonal kvantornak
  - Útvonalon értelmezett operátorok nem kombinálhatók

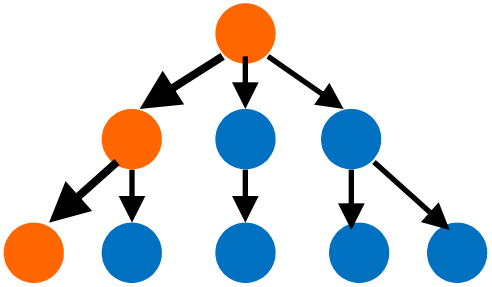
CTL operátorok (példák)



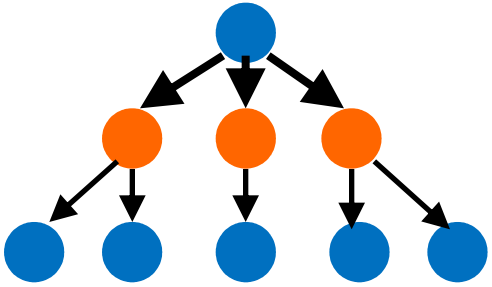
EX p



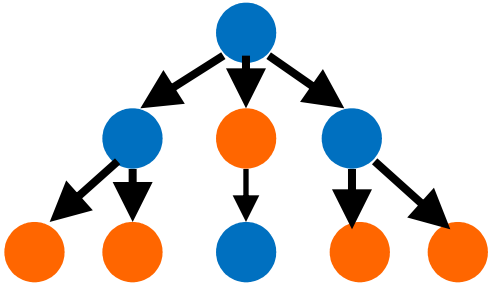
EF p



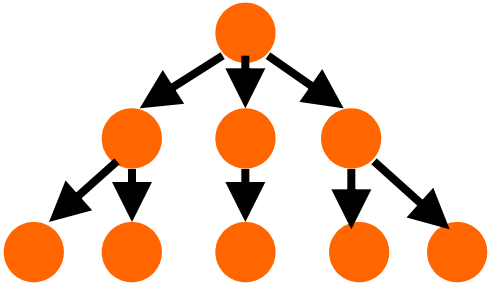
EG p



AX p



AF p



AG p

Példák CTL illetve CTL* kifejezésekre

- $A(p \Rightarrow F\ q)$:
  - Minden útvonalra igaz, hogy amennyiben p igaz az útvonal kezdetén, akkor ezt a jövőben olyan állapot fogja követni, ahol q igaz
- $E(p \wedge G\ q)$:
  - Létezik olyan útvonal, hogy ennek kezdetén p igaz és minden állapotán q is igaz
- $E(XXX\ p \vee F\ q)$:
  - Létezik olyan útvonal, hogy
    - vagy ennek negyedik állapotán igaz p,
    - vagy valamikor a jövőben igaz q
- AG EF p:
  - A rendszer bármelyik elérhető állapotára igaz, hogy onnan indulva olyan állapotba vihető a rendszer, ahol p igaz
- EF AG p:
  - Lehetséges, hogy a rendszer olyan állapotba kerül, hogy utána p mindig igaz lesz
- $AG\ (p \Rightarrow AF\ q)$:
  - A rendszer bármelyik elérhető állapotára igaz, hogy ha ott p igaz, akkor bármely onnan induló úton valamikor q mindenképpen igaz lesz

CTL* formális 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 fenti szabályok alapján generált állapot-kifejezések

„Kimaradt” operátorok

- $p \vee q$ jelentése $\neg((\neg p)\wedge(\neg q))$
  $p \Rightarrow q$ jelentése $(\neg p) \vee q$
  $p \equiv q$ jelentése $(p \Rightarrow q)\wedge(q \Rightarrow p)$
- true jelentése $p\vee(\neg p)$ (minden állapotra igaz),
  false jelentése $p\wedge(\neg p)$ (egy állapotra sem igaz)
- F p jelentése true U p
  G p jelentése $\neg F(\neg p)$

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 $\forall i \geq 0: (s_i, s_{i+1}) \in R$.
- $\pi^i=(s_i, s_{i+1}, s_{i+2},\ldots)$ a π útvonal szuffixe i-től.

- $M,\pi \models p$ jelentése:
  az M modellben a π útvonalon igaz p

- $M,s \models p$ jelentése:
  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**:
  - $M,\pi \models p$ (ahol 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

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

CTL formális szintaxis (CTL*-hoz képest)

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

Útvonal-kifejezések nem kombinálhatók, rögtön hozzájuk „ragad” az útvonal kvantor

CTL formális szemantika

- Állapot-kifejezések:
  - **S1**, **S2**, **S3** szabályok (lásd CTL*) változatlanok.
- Útvonal-kifejezések:
  - **P1**, **P2**, **P3** helyébe lépő új **P0** szabályra:

P0:

- $M,\pi \models X\ p$ ahol p állapot-kifejezés a.cs.a. $M,s_1 \models p$
- $M,\pi \models p\ U\ q$ ahol p,q állapot-kifejezés a.cs.a

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

Mintapélda: Kölcsönös kizárás

- 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 belépni akar: W1, W2
- Atomi kijelentések:
  AP = {C1, C2, N1, N2, W1, W2}

Mintapélda (folytatás)

- Biztonság: Egyszerre csak egy processz lehet a kritikus szakaszban:

  $AG\ (\neg(C1 \wedge C2))$
- Élőség: Ha egy processz várakozik, akkor előbb-utóbb mindig beléphet:

  $AG\ (W1 \Rightarrow AF(C1))$ illetve $AG\ (W2 \Rightarrow AF(C2))$
- Egy processz előbb-utóbb megpróbál belépni

  $AG\ (N1 \Rightarrow EF\ W1)$ illetve $AG\ (N2 \Rightarrow EF\ W2)$
- Előfordul, hogy ugyanaz a processz egymás után kétszer is belép a kritikus szakaszba:

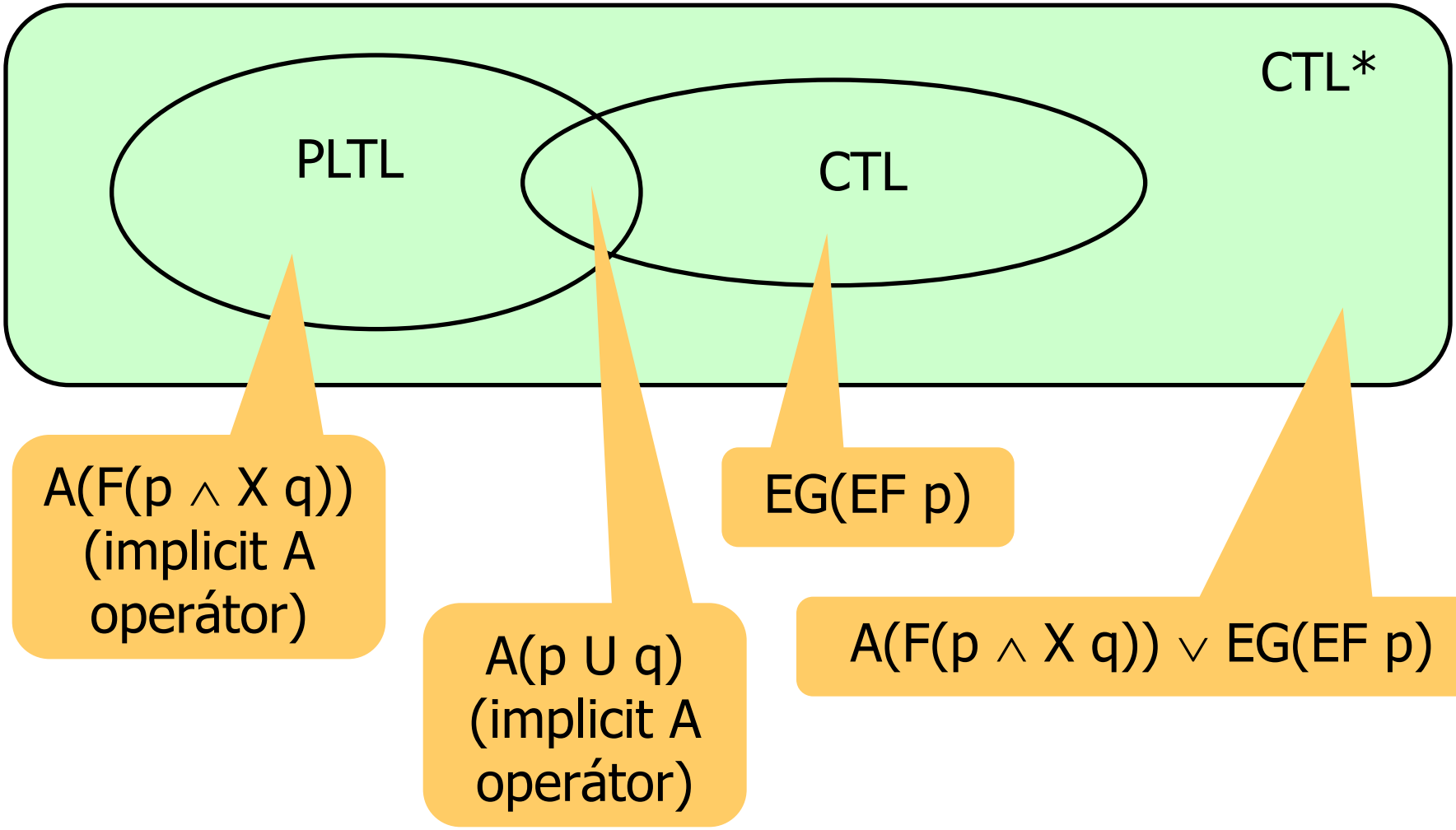
  $EF(C1 \wedge E\ [C1\ U\ (\neg C1 \wedge E[\neg C2\ U\ C1]\ )\ ])$ illetve

  $EF(C2 \wedge E\ [C2\ U\ (\neg C2 \wedge E[\neg C1\ U\ C2]\ )\ ])$

Kifejezőképesség

- Egy temporális logika kifejezőképessége nagyobb egy másikénál, ha képes olyan tulajdonság megfogalmazására, amire 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 PLTL kifejezéseket

LTL, CTL, CTL* kifejezőképessége



Kifejezőképesség - formálisan

- TL1 és TL2 logikák kifejezőképessége azonos, ha minden *M* modellre és annak minden *s* állapotára:

$$\forall p \in \mathrm{TL1}:$$

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

$$\forall q \in \mathrm{TL2}:$$

$$(\exists p \in \mathrm{TL1}:\ (M, s \models p \Leftrightarrow M, s \models q))$$

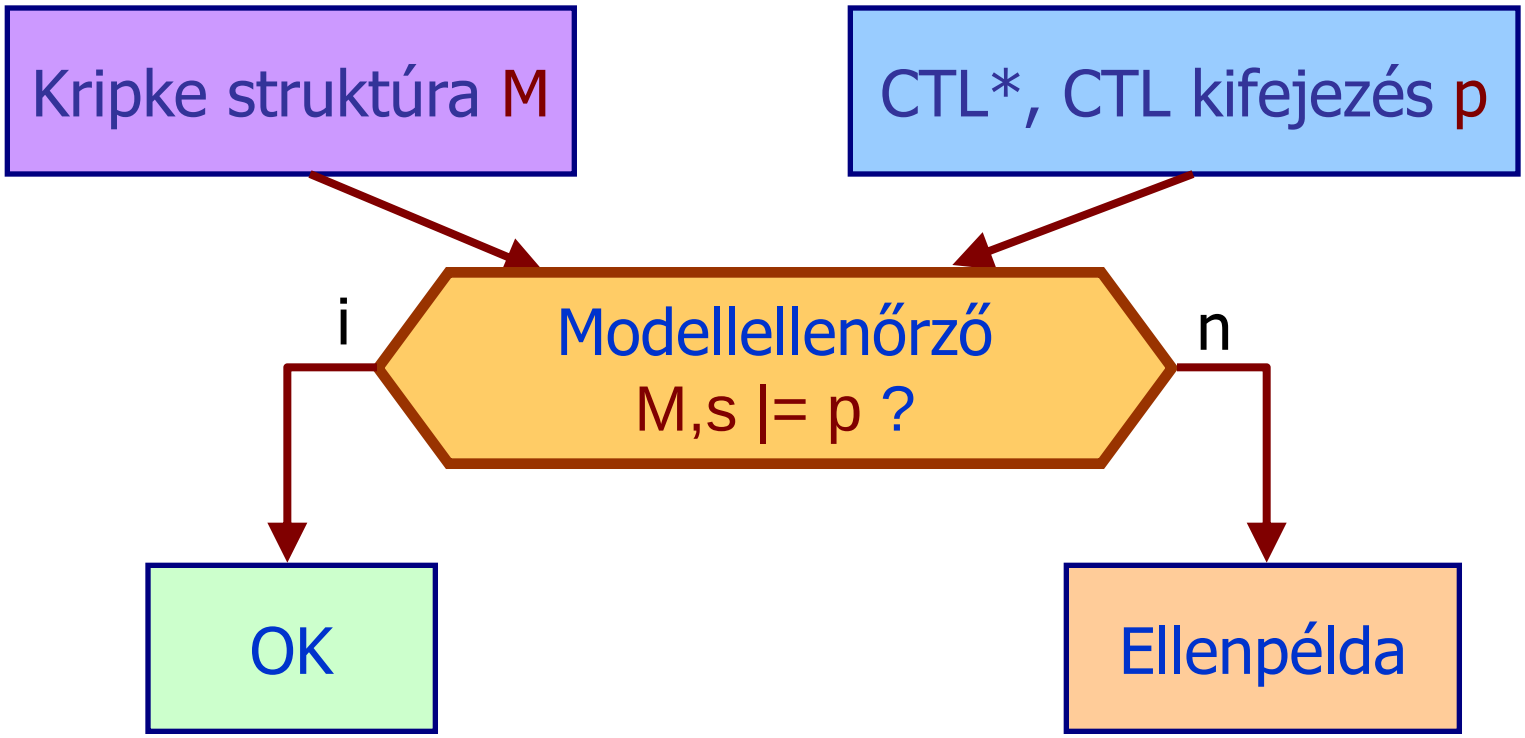
- Ha csak az első kifejezés igaz,
  akkor TL1 kevésbé kifejező, mint TL2

FairCTL: A „fair” utak megadása

- A követelmények ellenőrzésének korlátozása a „fair” útvonalakra
  - Triviális ellenpéldák kihagyása (pl. mindig reset, minden üzenet elveszik)
- Módosított kvantorok:
  - A_q : minden „fair” útvonalon
  - E_q : létezik „fair” útvonal
- Az A_q és E_q kvantorokban szereplő q útvonal-kifejezés (fair útvonal) formái:
  - GF r: az r állapot-kifejezés végtelen sokszor előfordulhat a „fair” útvonal mentén („nincs kiéheztetés”)
  - FG r: az r állapot-kifejezés csaknem mindig igaz a „fair” útvonal mentén („üzemi állapot beáll”)
- Módosított operátorok jelentése:
  - $A_qF\ p$ jelentése az $A(q \Rightarrow F\ p)$ CTL* kifejezés
  - $E_qG\ p$ jelentése az $E(q \wedge G\ p)$ CTL* kifejezés
- FairCTL előnyei:
  - „Fair” útvonalakra korlátozható az ellenőrzés
  - A CTL modellellenőrzés egyszerűsége megmarad

CTL modellellenőrzés: Szemantika alapon

A modellellenőrzés feladata

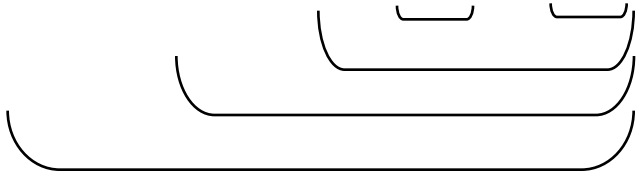


Megközelítés

- Globális modellellenőrzés:
  - Adott p CTL kifejezés esetén Sat(p) számítása: Azok az állapotok, ahol igaz p
  - Ezeket az állapotokat p-vel címkézzük
  - Ezután $s \in Sat(p)$ egyszerűen vizsgálható egy adott s (kezdő)állapotra
- Sat(p) számítása inkrementálisan történik
  - Címkézett állapothalmazok bővítése
  - A bővítés vége: Nem nő a címkézhető állapotok halmaza

CTL modellellenőrzés állapot címkézéssel

- Kifejezések felbontása azok struktúrája alapján, és „belülről kifelé” Sat(..) halmaz számítások:
  AF (P $\wedge$ E (Q U R))



- Állapotok címkézése: Hol igaz egy adott kifejezés?
  - Kiindulás: KS címkézve van atomi kijelentésekkel
  - Továbblépés: Címkézés az összetettebb kifejezésekkel
    - Ha p illetve q címkék már vannak, akkor megadható, hol lehet $\neg$p, p$\wedge$q, EX p, AX p, E(p U q), A(p U q) címke
    - Az eddigi címkéket alapul véve (adott szintaxis szabály szerint képzett) összetettebb kifejezésekkel való címkézés végezhető
    - Inkrementális címkézés a szemantika definíció alapján!

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-ban
- $\neg P$ 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) esetére kell még megadni a címkézési eljárást

Az AX, EX alakú kifejezések

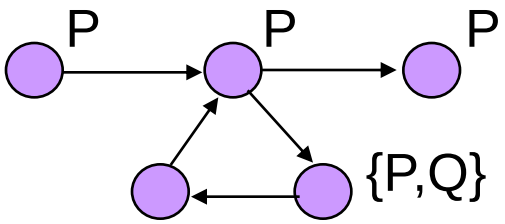
- EX p azokban az s állapotokban igaz, amelyeknek van olyan rákövetkező állapota, ahol p igaz
  - Egy állapot címkézése lehet EX p, ha van olyan rákövetkező állapota, ami p-vel címkézett

- 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

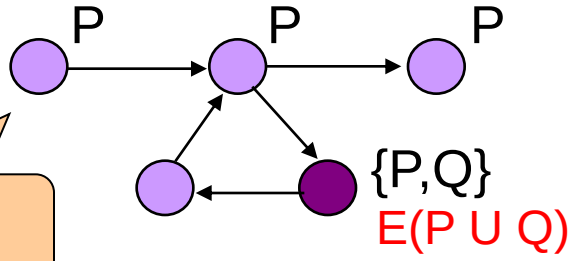
Az E(p U q) kifejezések

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

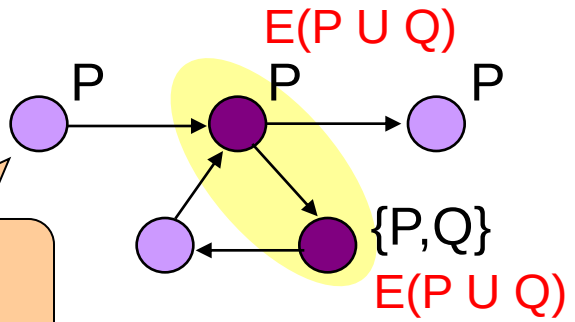
Az E(P U Q) címkézés



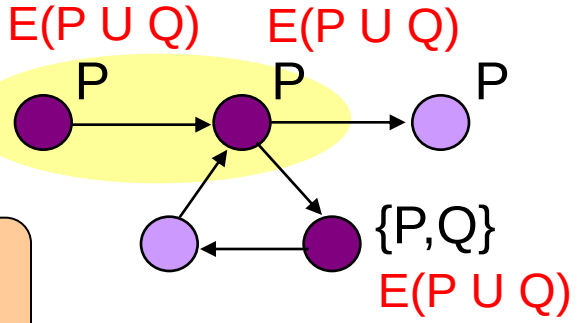
Kripke struktúra a kezdő címkézéssel



Első lépés: Q



Második lépés: P ∧ EX



Harmadik lépés: P ∧ EX

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

Az A(p U q) kifejezések

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

Ezzel a formális szintaxisban használt operátorokat lefedtük!

Címkézés leírása halmazműveletekkel

- A címkézés bővítése halmazműveletekkel történik
  - Z kezdőhalmaz adott (az előző lépések alapján)
  - Az inkrementális bővítés a már címkézett állapotok (legyen Z halmaz) megelőző állapotaira hivatkozik:

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

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

  Kérdés: $pre_A(Z)$ felírható-e $pre_E()$ segítségével?

- Példa: E(P U Q) esetén:
  - Kezdőhalmaz: $X_0 = \{s \mid Q \in L(s)\}$
  - Bővítés: $X_{i+1} = X_i \cup (pre_E(X_i) \cap \{s \mid P \in L(s)\})$

Eddig címkézettek és …

… ezek megelőző állapotai közül amelyek …

… P-vel címkézettek

  - Bővítés vége: Ha $X_{i+1} = X_i$, 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)
  - Lépések: Atomi kijelentésekből indítva az összetettebb kifejezések felé („belülről kifelé”)
  - Az előző lépésben adott címkézés felhasználása (virtuális „atomi kijelentések”)
- A(p U q), E (p U q) esetén: Inkrementális állapotcímkézés
  - Kezdőhalmaz:
    - A belső kifejezések (p, q) által meghatározott állapothalmaz alapján
  - Iteráció: A szemantika alapján (megelőző állapotokra lépegetve)
  - Iteráció vége: Nem nő a címkézett állapotok halmaza
    - Precíz matematikai algoritmus: Fixpont számítás teljes hálókban

CTL modellellenőrzés elméleti alapjai: Fixpont iteráció

Teljes hálók elmélete

- KS=(S,R,L) Kripke-struktúra
- $(2^S, \subseteq)$ teljes hálót képez, mivel
  - $\subseteq$ reflexív, tranzitív, antiszimmetrikus reláció
  - 2^S tartalmaz felső (S) és alsó ($\varnothing$) határt
- Legyen: τ: $2^S \rightarrow 2^S$ leképzés $\tau(z)$ jelöléssel
  - τ monoton, ha $z_1 \subseteq z_2$ esetén $\tau(z_1) \subseteq \tau(z_2)$
  - τ $\cup$-folytonos, ha $z_1 \subseteq z_2 \subseteq z_3 \subseteq \ldots$ esetén $\tau(\cup_i z_i) = \cup_i \tau(z_i)$
  - τ $\cap$-folytonos, ha $z_1 \supseteq z_2 \supseteq z_3 \supseteq \ldots$ esetén $\tau(\cap_i z_i) = \cap_i \tau(z_i)$
  - Ha S véges, akkor a monotonitásból adódik a $\cup$-folytonosság és $\cap$-folytonosság

Fixpontok

- Fixpontok definíciója:
  - lfp $\tau(z)$ az a legkisebb $z \subseteq S$, amelyre $\tau(z)=z$
  - gfp $\tau(z)$ az a legnagyobb $z \subseteq S$, amelyre $\tau(z)=z$
- Tételek:
  - Tarski tétele (létezés):
    Ha S véges és $\tau(z)$ monoton, akkor létezik legkisebb és legnagyobb fixpont
  - Kleene tételei (számítási mód):
    Ha $\tau(z)$ $\cup$-folytonos, akkor lfp $\tau(z)= \cup_i \tau^i(\emptyset)$.
    Ha $\tau(z)$ $\cap$ -folytonos, akkor gfp $\tau(z)= \cap_i \tau^i(S)$.

Fixpontok számítása

- A tételek alapján belátható következmény:
  - Véges S és monoton $\tau(z)$ esetén:

$\exists\, i_0$: lfp $\tau(z) = \tau^{i_0}(\varnothing)$, ld. lfp $\tau(z) = \cup_i \tau^i(\varnothing)$

$\exists\, j_0$: gfp $\tau(z) = \tau^{j_0}(S)$, ld. gfp $\tau(z) = \cap_i \tau^i(S)$

$$\begin{array}{ccccc} \varnothing = z_0 & \subseteq & z_1 & \subseteq & z_2 \\ & & \| & & \| \quad \ldots \\ & & \tau(z_0) & \subseteq & \tau(z_1) \end{array}$$

Fixpontok számítása

- A tételek alapján belátható következmény:
  - Véges S és monoton $\tau(z)$ esetén:

    $\exists\, i_0$: lfp $\tau(z) = \tau^{i_0}(\varnothing)$, ld. lfp $\tau(z) = \cup_i \tau^i(\varnothing)$

    $\exists\, j_0$: gfp $\tau(z) = \tau^{j_0}(S)$, ld. gfp $\tau(z) = \cap_i \tau^i(S)$

$$\begin{array}{ccccc} \varnothing = z_0 & \subseteq & z_1 & \subseteq & z_2 \\ & & \| & & \| \quad \ldots \\ & & \tau(z_0) & \subseteq & \tau(z_1) \end{array}$$

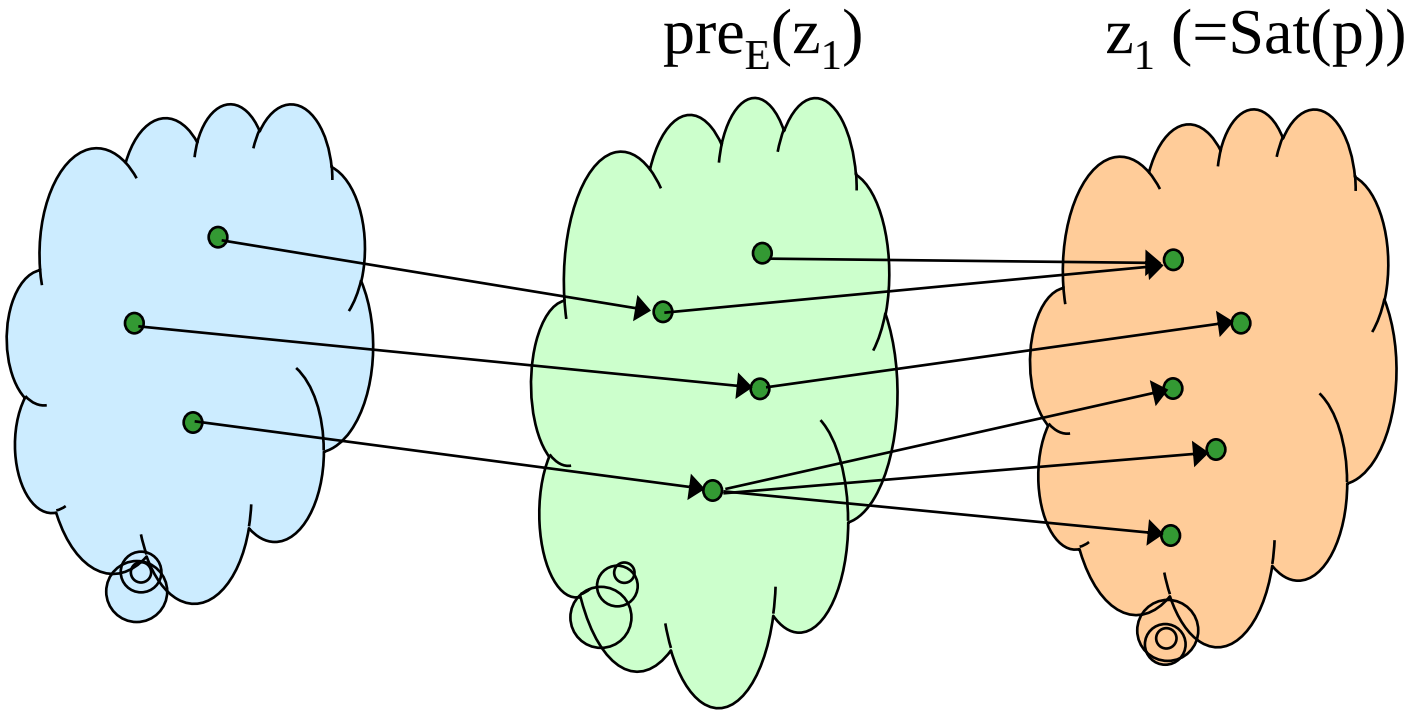
- Iterációval való számítás (korlát az S mérete):
  - lfp $\tau(z)$: $\varnothing, \tau(\varnothing), \tau^2(\varnothing), \ldots$ amíg $\tau^i(\varnothing) = \tau^{i+1}(\varnothing)$
    - Monoton nő a halmaz mérete a fixpontig
  - gfp $\tau(z)$: $S, \tau(S), \tau^2(S), \ldots$ amíg $\tau^j(S) = \tau^{j+1}(S)$
    - Monoton csökken a halmaz mérete a fixpontig

CTL operátorok és fixpont műveletek

- Pongyola megfogalmazásban:
  - lfp: eshetőségek (pl. EF); adott tulajdonságot teljesítő állapotokhoz vezető utak kiinduló állapotai
  - gfp: folyamatos igazságok (pl. EG): adott tulajdonságot folyamatosan teljesítő utak kiinduló állapotai

- Bizonyítható tétel: $Sat(EF\ p) = lfp\ \tau(z)$,
  - ahol $\tau(z) = Sat(p) \cup pre_E(z)$ itt analógia: $EF(p) = p \vee EX\ EF(p)$
  - ahol $pre_E(z) = \{s \mid \exists t: (s,t) \in R \text{ és } t \in z\}$, azaz azoknak az állapotoknak a halmaza, ahonnan van átmenet z-be
  - Itt fixpont számítás: „visszafelé lépked az utakon”; kezdőállapotokat keres a p-t kielégítő állapotba vezető utakhoz
    - Kezdetben $\varnothing$, ebből Sat(p) lesz az első bővítés
    - Visszalépés az odavezető állapotokon: $pre_E(z)$ lépked visszafelé

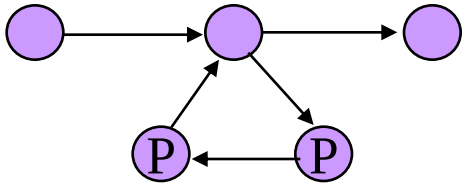
$pre_E(z)$ számítása



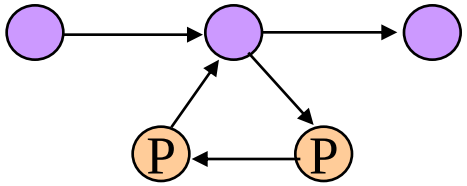
- Sat(p) lesz az első lépés eredménye
- $pre_E(z)$ az iteráció során visszafelé lépked az egyes utakon, kezdőállapotokat keresve a Sat(p)-be vezető utakhoz

EF(P) iterációs példa

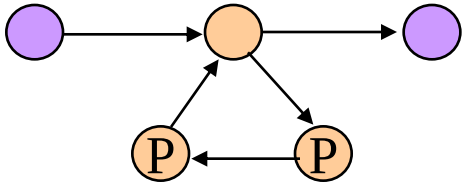
$z_0 = \varnothing$



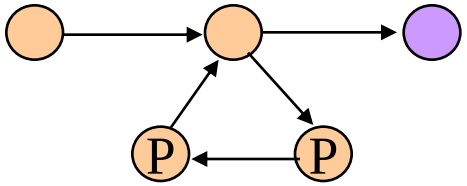
$z_1 = Sat(P) \cup pre_E(z_0) = Sat(P)$



$z_2 = Sat(P) \cup pre_E(z_1)$



$z_3 = Sat(P) \cup pre_E(z_2)$



CTL operátorok és fixpont műveletek

Tételek (folytatás):

Terminál utakon sántít az analógia

- Sat(EG p)= gfp $\tau(z)$,
  - ahol $\tau(z)$= Sat(p) $\cap$ $pre_E(z)$

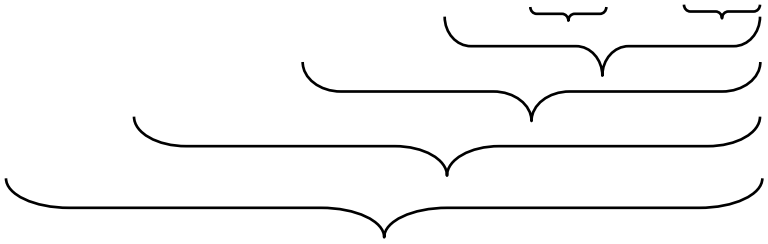
    analógia: $EG(p)=p \wedge EX\ EG(p)$
  - Itt fixpont számítás: csak az olyan utak kellenek, ahol p minden állapotban igaz (ld. metszet Sat(p)-vel)
    - Kezdetben S, ebből az elérhető Sat(p)
    - Visszalépés az olyan odavezető állapotokon, ahol igaz p

- Sat(E(p U q))= lfp $\tau(z)$,
  - ahol $\tau(z)$= Sat(q) $\cup$ (Sat(p) $\cap$ $pre_E(z)$)

    analógia: $E(pUq)=q \vee (p \wedge EX\ E(pUq)$

Összefoglalás: CTL modellellenőrzés iteratívan

- Alapelv: Kifejezések felbontása és „belülről kifelé” Sat(..) számítások:
  AX AG (p $\wedge$ E (Q U R))



- Szabályok: A szintaxis definíció alapján
  - Sat(P) = {s | P$\in$L(s)}
  - Sat($\neg$p) = S \ Sat(p)
  - Sat(p$\wedge$q) = Sat(p) $\cap$ Sat(q)
    Sat(p$\vee$q) = Sat(p) $\cup$ Sat(q)

CTL modellellenőrzés iteratívan (folytatás)

- Szabályok (folytatás):
  - Sat(EF p) számítás: lfp $\tau(z)$
    - ahol $\tau(z)= Sat(p) \cup pre_E(z)$
    - ahol $pre_E(z)=\{s \mid \exists t: (s,t)\in R \text{ és } t\in z\}$

    azaz az iteráció:
    - $z_0= \varnothing$
    - $z_1= \tau(z_0)=\tau(\varnothing)$
    - $z_{i+1}= \tau(z_i)= Sat(p) \cup pre_E(z_i) = Sat(p) \cup \{s \mid \exists t: (s,t)\in R \text{ és } t\in z_i\}$
    - amíg $z_{i+1}= z_i$ és itt z_i = lfp $\tau(z)$ = Sat(EF p)

CTL modellellenőrzés iteratívan (folytatás)

- Szabályok (folytatás):
  - Sat(EG p) számítás: gfp $\tau(z)$
    - ahol $\tau(z)$= Sat(p) $\cap$ $pre_E(z)$
    - ahol $pre_E(z)=\{s \mid \exists t: (s,t)\in R$ és $t\in z\}$

    azaz az iteráció:
    - z_0= S
    - $z_1= \tau(z_0)=\tau(S)$
    - $z_{i+1}= \tau(z_i)$= Sat(p) $\cap$ $\{s \mid \exists t: (s,t)\in R$ és $t\in z_i\}$
    - amíg $z_{i+1}= z_i$ és itt z_i = gfp $\tau(z)$ = Sat(EG p)
  - Sat(E(p U q)) számítás hasonlóan
  - További CTL operátorok mint helyettesítések kezelhetők
- A modellellenőrzés halmazműveleteket jelent!

CTL modellellenőrzés komplexitása

- Worst case időkomplexitás: $O(|S|^2 \times |p|)$
  - $|S|^2$ az átmenetek száma worst case esetben (ld. pre(z))
  - $|p|$ az operátorok száma;
    itt a szintaxis szabályainak megfelelően bontjuk fel a kifejezést és külön-külön számítjuk a Sat(..) halmazt
  - PLTL-nél kedvezőbb (pedig elágazó idejű logika);
    - De itt a PLTL kifejezés soha nem hosszabb, mint az ugyanazt a tulajdonságot leíró CTL kifejezés (ha van ilyen)
- FairCTL esetén: $O(|S|^2 \times |p| \times |q|)$
  - Itt q a fair útvonalak megadására szolgáló kifejezés
- CTL* esetén: $O(|S|^2 \times 2^{|p|})$
  - Itt a beágyazott PLTL kifejezések kötetlensége miatt jön be komplexitásnövekedés
  - ld. PLTL modellellenőrzés komplexitása: $O(|S|^2 \times 2^{|p|})$

Modális mu-kalkulus (kitekintés)

Modális mu-kalkulus

- Szintaxis KTS-en:

$$p ::= P \mid Z \mid \neg p \mid p \wedge p \mid [a]p \mid \langle a \rangle p \mid \mu Z.p \mid \nu Z.p$$

- Közvetlenül a fixpont operátorokat tartalmazza
  - $\nu Z.p$ legnagyobb fixpont (Z változó, ahol p nyitott kifejezés Z-re)
    - Az a legnagyobb $S^* \subseteq S$ halmaz, melyet újra megkapunk, ha a p(Z) formulát azzal az interpretációval értékeljük ki, hogy Z az S^*-on igaz
  - $\mu Z.p$ legkisebb fixpont (ahol p nyitott kifejezés Z-re)
- Előírás: Z páros számú negáció hatókörében fordulhat elő
  - Így kapunk monoton kifejezéseket, Sat(p) iterációval számítható
- CTL* is lefedhető vele (kifejezőképessége nagyobb)
- Worst case időkomplexitás: $O(|S|^2 \times |p|^a)$
  - Itt a az egymásba ágyazott váltakozó fixpont operátorok száma („alternation depth”)
  - CTL esetén az alternation depth értéke 1 (független fixpont műveletek), általános esetben ennél nagyobb is lehet

A modális mu-kalkulus használata

- CTL: egymásba ágyazott fixpont kifejezések között nincs függőség:
  pl. AG EF p = $\nu Z.(\mu Y.(p \vee EX(Y)) \wedge AX(Z))$
  - Egy belső fixpont kifejezés nem függ egy külső fixpont kifejezés változóitól
  - Egy-egy kifejezés kiértékelhető belülről kifelé haladva, egyenként feloldva az egyes operátorokat
- Általános eset: Lehet függőség
  pl. $\nu Z.\mu Y.(\langle b \rangle Z \vee \langle a \rangle Y)$, azaz van a és b lépésekből álló út, végtelen sokszor előforduló b-vel
  - Kölcsönös függőség van a legkisebb és legnagyobb fixpont kifejezések között
  - Az iteráció ”ellenkező irányba tart”, újraszámolt belső iteráció szükséges a külső iterációs lépésekben