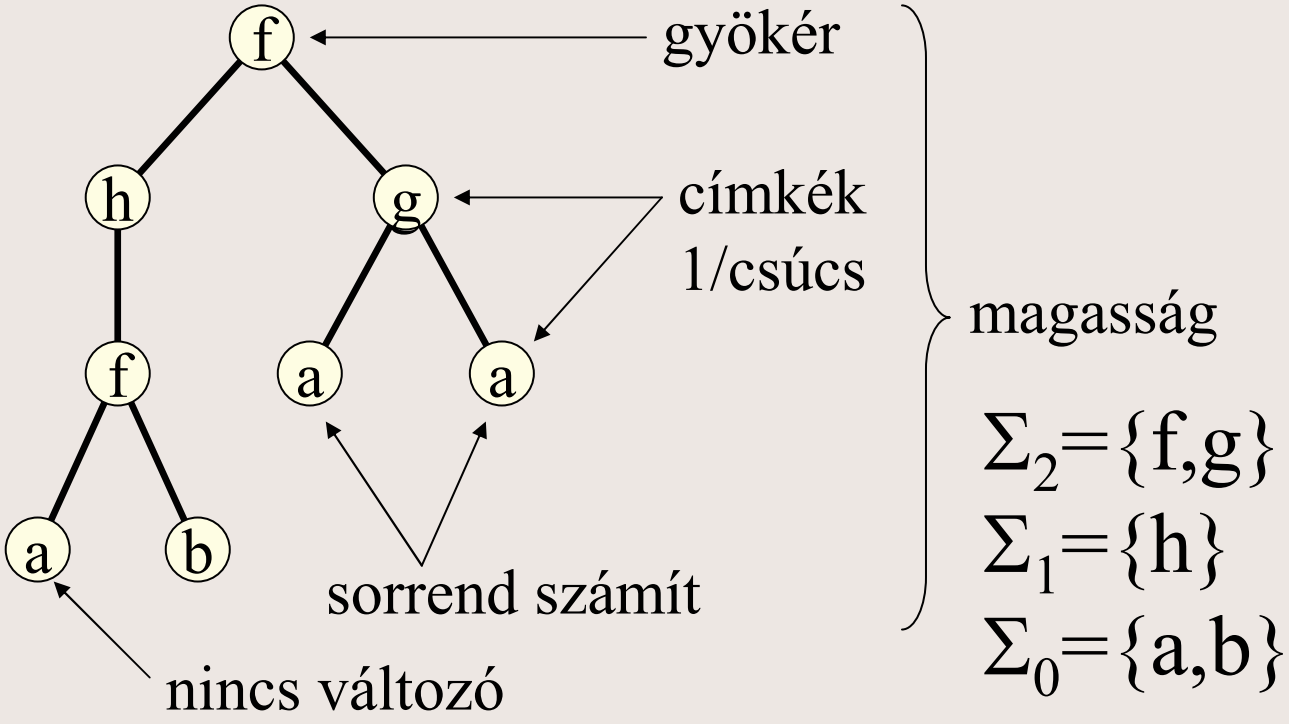
Logikák véges fákon

Iván Szabolcs
Szegedi Tudományegyetem
Számítástudomány Alapjai Tanszék

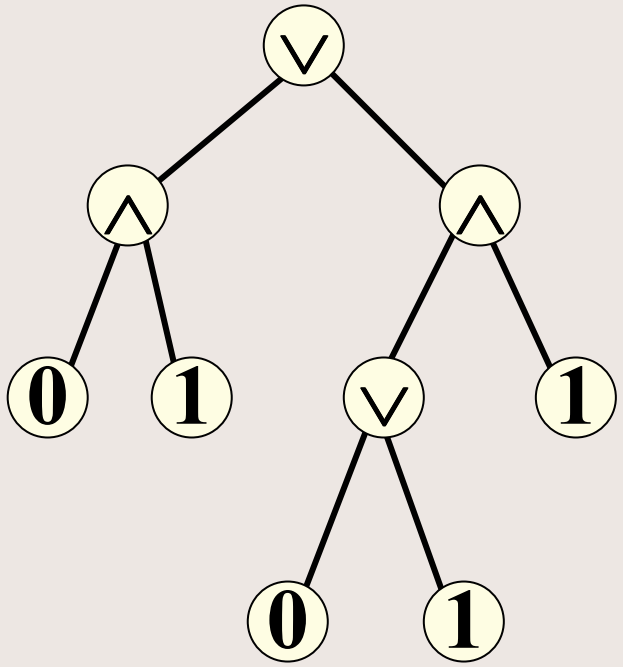
Fák

- Σ véges rangolt ábécé



Fanyelvek

- Legyen $\Sigma_2=\{\wedge,\vee\}$ és $\Sigma_0=\{\mathbf{0},\mathbf{1}\}$



…és a többi
1 értékű fa

Faautomaták

- Faautomata: egy Σ-algebra, elemeinek egy kitüntetett F „végállapot”-halmazával
- Felismer egy fát, ha azt kiértékelve az eredmény F-beli
- Az L fanyelv *reguláris*, ha van hozzá faautomata

Faautomata példa

- Megint az **1** értékű fák:

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

$\mathbf{0} \rightarrow 0$

$\mathbf{1} \rightarrow 1$

$F=\{1\}$

Párhuzam: szavak

- A szavak „valójában” fák
- A klasszikus automaták speciális faautomaták

$aab \equiv$ 

$\Sigma_0 = \{\lambda\}$, értéke q_0
Σ_1: az input ábécé,
megfelel az átmeneteknek

Lekérdezések

- n-változós lekérdezés (query): fákhoz n-változós relációkat rendel
  - A reláció V^n egy részhalmaza; itt V a fa csúcsainak halmaza
- 0-változós (boolean) lekérdezés: minden fára vagy igaz, vagy hamis
  - Vagyis fanyelv
  - Automata csak ilyet tud

Lekérdezések II.

- 1-változós lekérdezés: a fa bizonyos csúcsait adja vissza
  - Ilyenek az XML Xpath kérései
  - Manapság nagy rájuk az igény
- 2-változós lekérdezés: csúcspárok
  - Mint pl. a „gyereke” tranzitív lezártja

Logikák

Formalizmusok lekérdezésekre

- Változót tartalmazó logikák
  - MSO
  - FO és variánsai
- Változómentes logikák
  - CTL* és fragmensei

Monadic Second Order

Szintaxis:

- $F \rightarrow p_a(x) \mid x <_{ch} y \mid x <_{sb} y \mid x \in X \mid x = y$
- $F \rightarrow F \vee F \mid \neg F \mid \exists x.F \mid \exists X.F$

Szemantika:

- x egy csúcs
- X csúcsok egy halmaza
- $p_a(x)$: x címkéje a
- $<_{ch}$: gyerek $<_{sb}$: jobbtestvér

MSO példák

$x <^{+}_{ch} y$:

$$\exists X.(x \in X \wedge (\forall z.z \in X \rightarrow (\exists w.z <_{ch} w \wedge (w \in X) \vee (w = y))))$$

$$Root(x) : \forall y.\neg(y <_{ch} x) \qquad Leaf(x) : \forall y.\neg(x <_{ch} y)$$

Sőt:

Régi ismerős

$$\exists X.(\ \forall x.\ x \in X \leftrightarrow (\ p_1(x) \vee p_{\wedge}(x) \wedge (\forall y.x <_{ch} y \rightarrow y \in X) \vee p_{\vee}(x) \wedge (\exists y.x <_{ch} y \wedge y \in X))) \wedge \forall x.\ Root(x) \rightarrow x \in X\ )$$

A gyanú helyes

- Minden reguláris fanyelvhez van boolean MSO lekérdezés
  - Sőt $\exists$MSO
- Minden boolean MSO lekérdezés reguláris fanyelvet ad meg
  - Konstruktív, de nem elemi!

Párhuzam: szavak

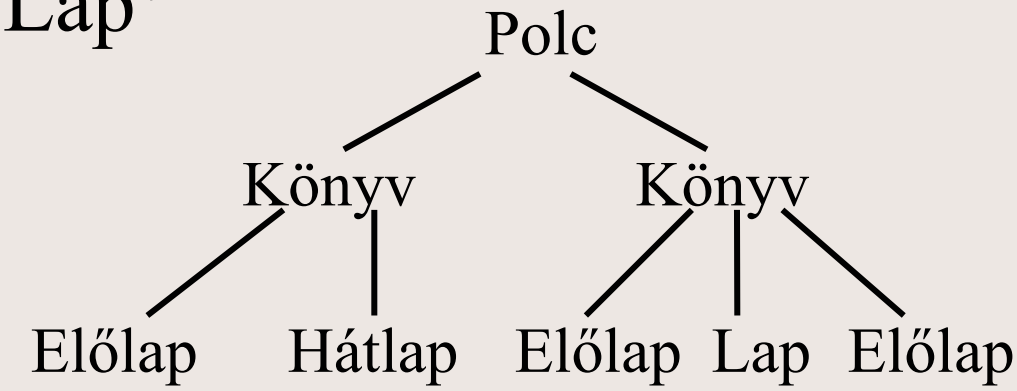
- „A következő három nyelvosztály megegyezik:
  - A reguláris nyelveké
  - Az MSO mondattal kifejezhetőké
  - Az ∃MSO mondattal kifejezhetőké”
- Ez tehát fákra is így van
  - a bizonyítás is analóg

Az XML EDTD modellje

- Az XML DTD-k felfoghatók egy-egy módosított CFG-ként:
  - Olyan (Σ',S,P,*h*) négyes, ahol a P átírási halmaz $a \rightarrow e$ alakú szabályokból áll
  - itt $a \in \Sigma'$, e pedig Σ' fölötti reguláris kifejezés
  - *h:* $(\Sigma')^* \rightarrow (\Sigma)^*$ homomorfizmus (technikai okok miatt)
  - De itt az ábécé nem rangolt
  - EDTD: Extended DTD, zártság miatt kell

Példa

- Könyvespolc $\rightarrow$ (Könyv + Folyóirat)$^+$
- Könyv $\rightarrow$ Előlap Lap* Hátlap
- Folyóirat $\rightarrow$ Lap$^+$
- többi $\rightarrow \lambda$
- *h* most *id*



EDTD $\subseteq$ MSO

- Minden EDTD-hez készíthető MSO formula
  - Automatához hasonló módon
  - Először címkénként szétparticionáljuk
  - Azután a gyerekek listáján végighúzunk egy megfelelő stringautomatát (ez megy MSO-val)

Nincs új a nap alatt

- Az MSO mondatokkal definiálható fanyelvek osztálya egybeesik az EDTD-vel definiálható (nem-rangolt) fanyelvekével

Csakhogy…

- Egy F MSO mondatra és egy T fára azt eldönteni, hogy T modellje-e F-nek, *nagyon* nehéz
  - Vagy **nincs** olyan elemi f függvény, amire $O(f(|F|)|T|)$ idő elég lenne…
  - …vagy $P=NP$.
- Rögzített F-re egyébként elég $O(|T|)$ idő
- Kompromisszum: ETL (Efficient TL)

Efficient Tree Logic

- Cél: MSO-val megegyező kifejezőerő, jobb model-checking korláttal

Szintaxis (részletek is mindjárt):

$$F \to p_a(x) \mid x <_{ch} y \mid x <^{*}_{ch} y \mid x \in X \mid x = y$$

$$F \to F \vee F \mid \neg F$$

$$F \to \exists y.(x <_{ch} y \wedge F) \mid \exists y.(x <^{*}_{ch} y \wedge F)$$

$$F \to \exists X.((\forall y.y \in X \to x <^{*}_{ch} y) \wedge F)$$

} „őrök”

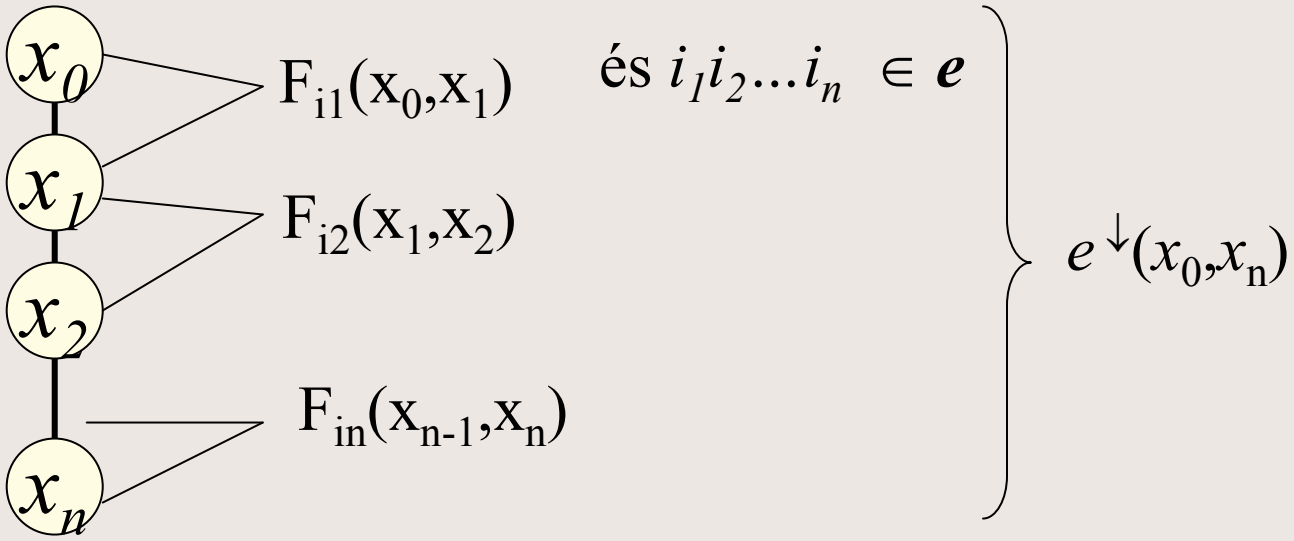
$$F \to e^{\downarrow}(x, y) \mid e^{\rightarrow}(x, X)$$

ETL vs. MSO

- Egyrészt kvantifikálni csak *őrzött* (guarded) módon engedélyezett
- Másrészt nem használunk $<_{sb}$ relációt
- Harmadrészt vannak az *útformulák:*
  - $e^{\downarrow}(x,y)$: vertikális útformula
  - $e^{\rightarrow}(x,X)$: horizontális útformula
- Egyéb szintaktikus megszorítások is vannak (a szabad változók halmazára)

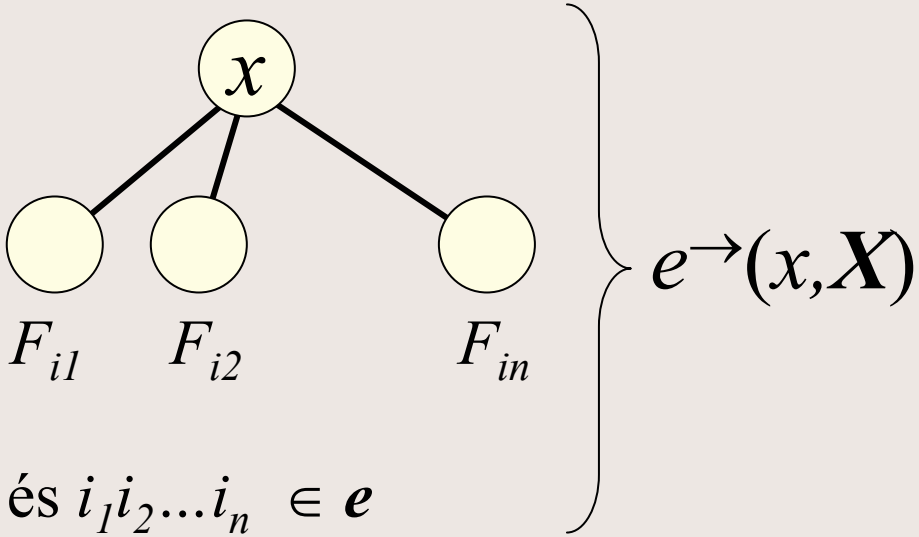
Vertikális útformulák

- Legyenek $F_1(x,y)$, $F_2(x,y)$, ..., $F_k(x,y)$ ETL formulák és e egy reguláris kifejezés $\{1,...,k\}$ fölött



Horizontális útformulák

- $F_1(x,\boldsymbol{X}), \ldots, F_k(x,\boldsymbol{X})$ ETL formulák, e egy reguláris kifejezés $\{1,\ldots,k\}$ fölött



ETL $\subseteq$ MSO

- Egyedül a kétféle útformula szorul magyarázatra
- De azokban <-utak regularitását és (indukció szerint) MSO teszteket kérjük
- Vagyis végig lehet húzni rajtuk egy megfelelő automatát (ami pedig MSO stringeken)

MSO $\subseteq$ ETL

- A (regularitás miatt létező) faautomatából indulva már majdnem ETL formulát is kapunk
- Csak annyi a gond, hogy sok lesz a „párhuzamosan szabad” halmazváltozó
- Ezeket trükkösen lecserélhetjük egyetlen halmazváltozóra
  - A halmaz elemei olyan csúcsok, amik körüli adott $(2N+1)$ sugarú környezetek lefedik a fát
  - Ezekre a környezetekre útformulákat alkalmazunk

ETL model checking

- ETL^o: csak olyan ETL formulák, amik diszjunktív normálformájúk
  - Nyilván ETL ekvivalens ETL^o-val
- Model-checking bonyolultság ezekre:
  - ETL^o-ra $2^{O(|F|)}|T|$
  - ETL-re emiatt duplán exponenciális már elég
  - Nem ismert, hogy ez éles-e
- És: ETL és MSO kifejezőereje megegyezik a boolean és unáris lekérdezéseken!

First Order Logic

Szintaxis

- $F \rightarrow p_a(x) \mid x <_{ch} y \mid x <_{sb} y \mid x = y$
- $F \rightarrow F \vee F \mid \neg F \mid \exists x.F$

Szemantika

- MSO megfelelő része

FO vs. FO(<)

- Láttuk, hogy MSO-ban kifejezhető <-ből $<^*$ és viszont
- FO-ban azonban csak $<^*$-ból fejezhető ki <, a másik irány nem oldható meg
- FO(<) az az elsőrendű logika, ahol < helyett $<^*$ az alapformulák eleme
- FO(<) ezért kifejezőbb, mint FO

A szavak esete

- Szavak esetében tudjuk, hogy az FO(<) definiálhatósági problémája eldönthető
  - Az L string nyelv pontosan akkor definiálható FO(<)-ben, ha szintaktikus monoidja aperiodikus
  - Egyébként FO(<) stringekre ekvivalens az LTL logikával (ezt később tárgyaljuk)
  - Sőt: ez egybeesik a csillagmentességgel is
  - PSPACE-teljes kérdés

Különbség

- Fákra jelenleg **nem ismert**, hogy FO(<) definiálhatósági problémája eldönthető-e vagy sem
  - Annyit tudunk, hogy FO(<) bináris fákra ekvivalens CTL*-gal (amit később tárgyalunk)
  - Itt FO(<) gyengébb, mint a csillagmentesség

FO(<) és MSO között: MPL

MPL: Monadic Path Logic

- Szintaxisa ugyanaz, mint MSO-é, a relációk tranzitív változatával
- Szemantikája azonban eltér: az X változók csak olyan csúcshalmazokat vehetnek fel értékül, melyek egy *teljes gyökér-levél utat* határoznak meg
- Ez hasonlít a temporális útformulákra

FO(<) és MSO között?

- Az világos, hogy minden FO(<)-formulához létezik ekvivalens MPL formula
- Visszafelé is van átalakítás:

$$\exists X.F \mapsto \exists t.(\mathrm{Leaf}(t) \wedge F')$$

$$x \in X \mapsto x <_{ch}^{*} t$$

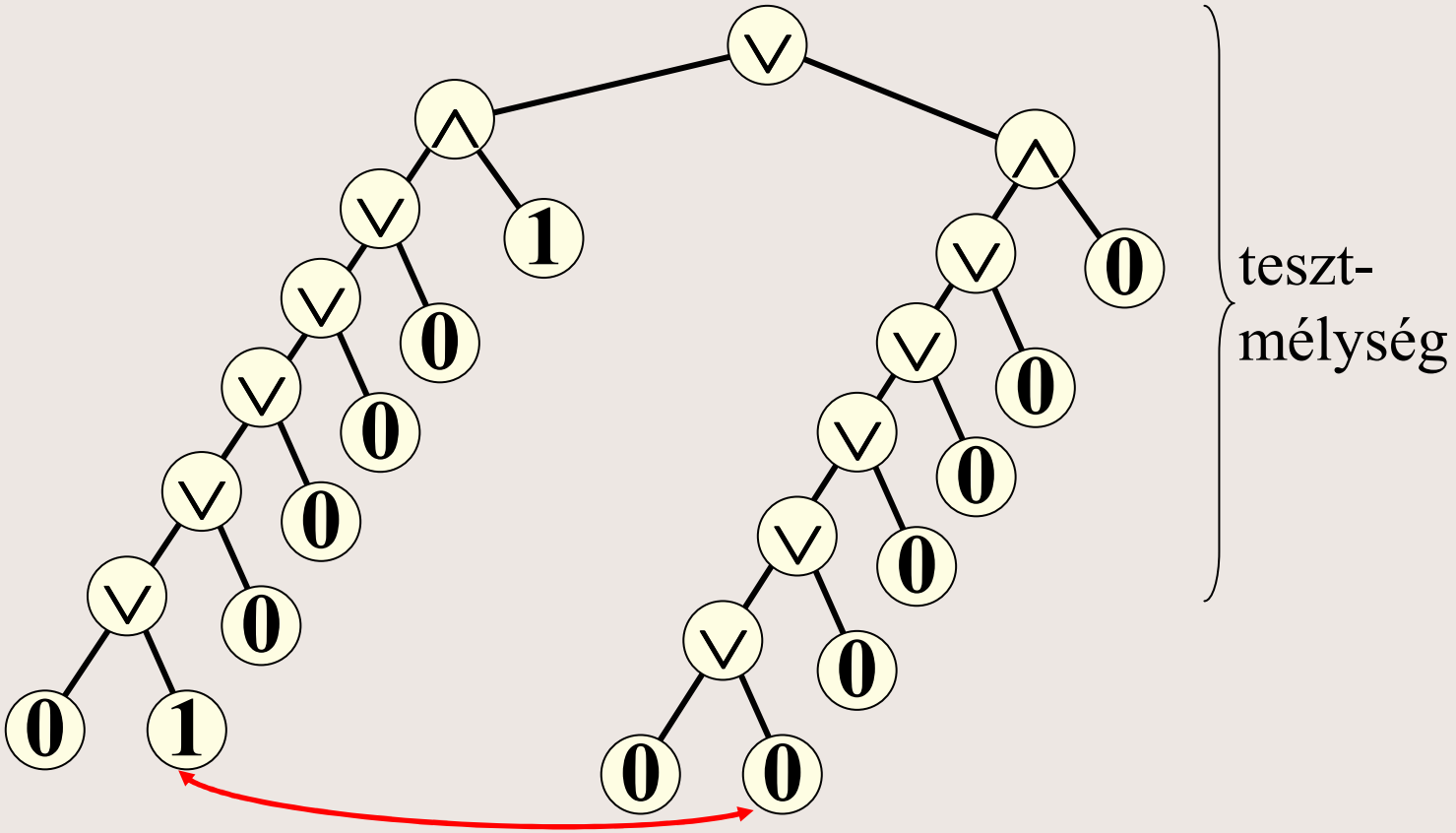
- Tehát MPL nem más, mint FO(<)

Lokális küszöbtesztelés

- Adott d és k egészekre és T_1, T_2 fákra fennáll a $T_1 \equiv_{d,k} T_2$ reláció, ha véve bármely legfeljebb d magasságú t „fát”:
  - mindkét fában ugyanannyiszor fordul elő t
  - vagy mindkét fában legalább k-szor
- Az L fanyelv *lokálisan küszöbtesztelhető* (LTT - locally threshold testable), ha valamely d-re és k-ra $\equiv_{d,k}$ szaturálja

LTT példa

- Kedvenc nyelvünk pl. *nem* LTT:



FO és LTT?

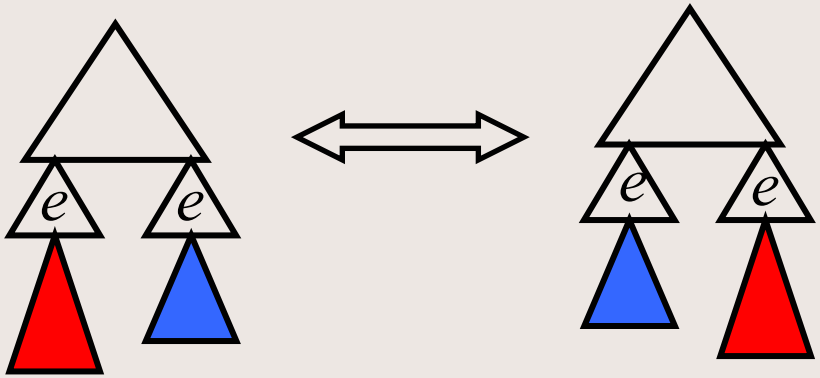
- Az LTT nyelvek FO-definiálhatóak
  - FO-ban tudunk beszélni véges fákról és „max *k* különböző” ill. „legalább *k* különböző” formulákat is tehetünk
- Az FO-definiálható nyelvek LTT nyelvek
  - Belátható, hogy ha két csúcs távolsága a Gaifman-gráfban több, mint 3^k, akkor legalább *k* további változó kell őket összekötni

FO és LTT: párhuzam

- Stringekre is: egy fanyelv pontosan akkor definiálható FO-ban, ha lokálisan küszöbtesztelhető
- Ez még önmagában nem ad eldöntési algoritmust

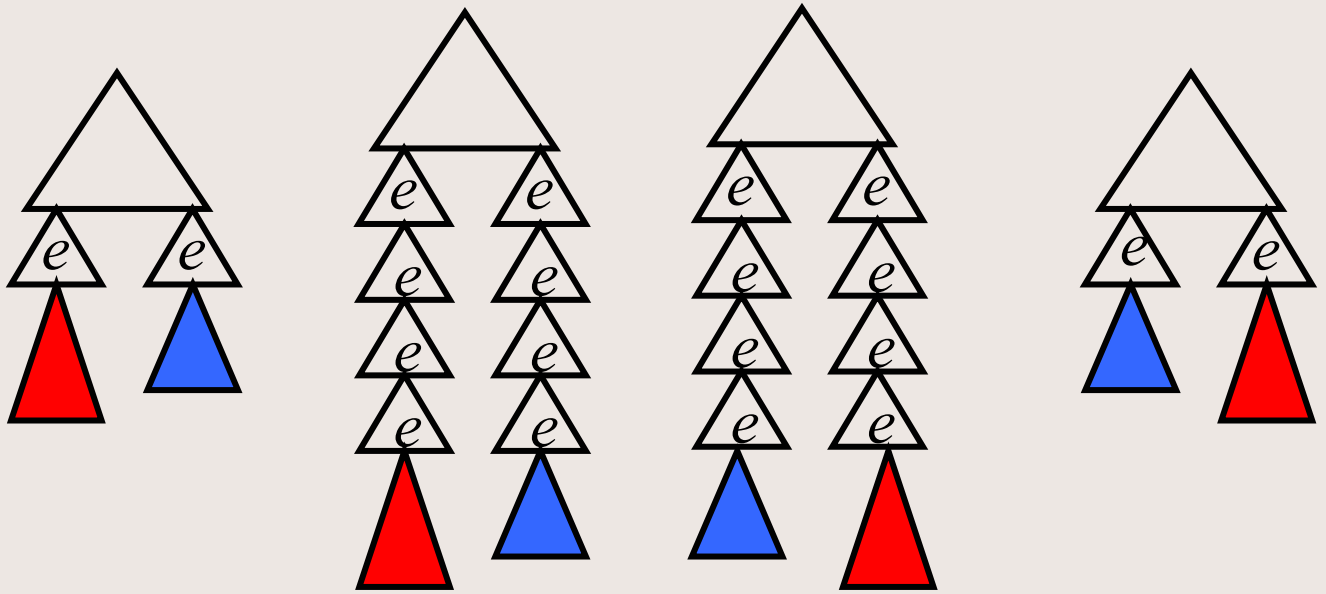
FO (azaz LTT) eldöntése

- T_0: fák; T_1: contextek; T_2: két változó
- $e \in T_1$ *idempotens,* ha $e^A = ee^A$
- Széltében csere (ilyen stringeken nincs):



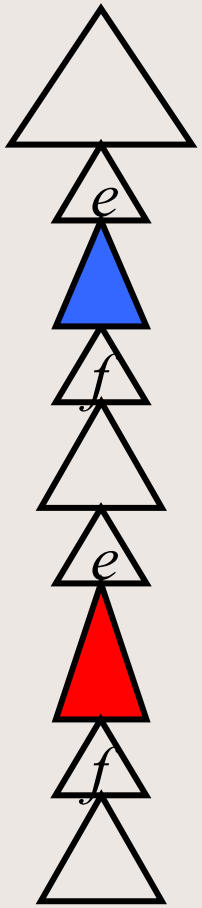
Széltében csere

- Ha L FO-definiálható, akkor invariáns a széltében cserére



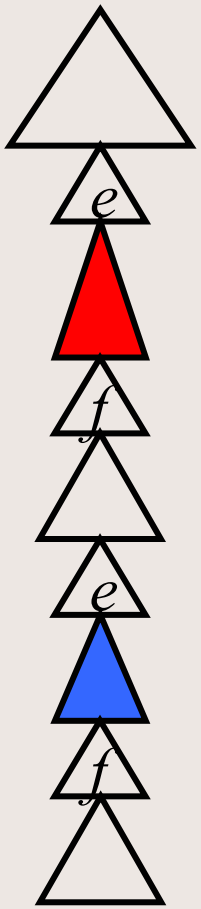
idempotens lokalitás idempotens

Hosszanti csere



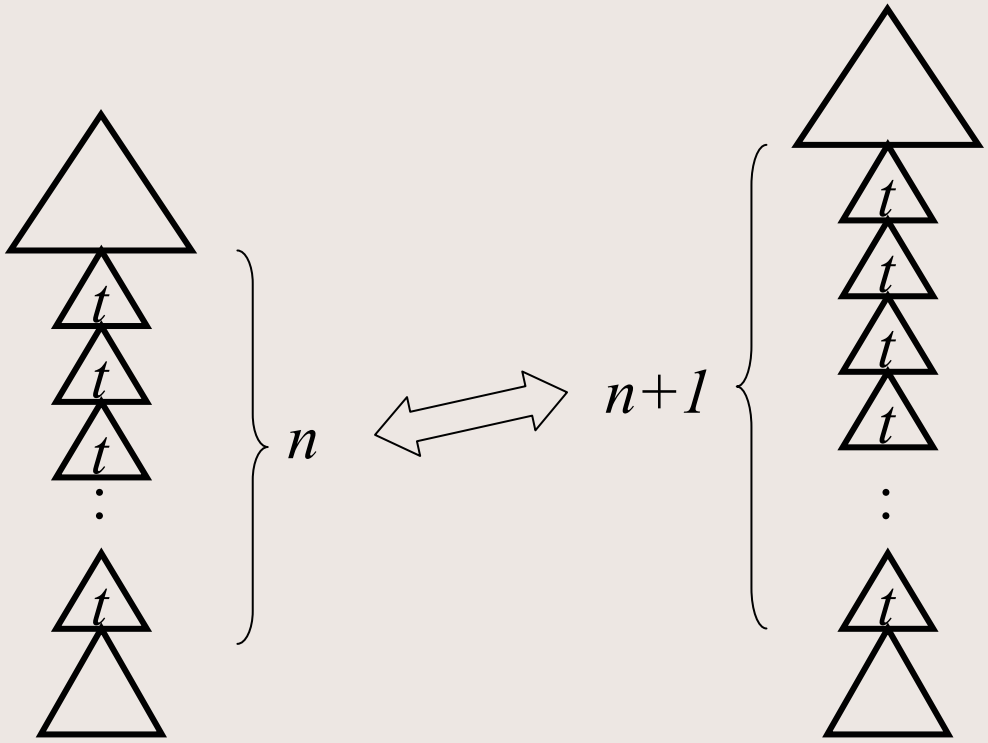
polcolhatunk
e-t és *f*-et
ismét

Erre is invariáns



Aperiodicitás

- Létezik *n*, amire



LTT miatt ez is stimmel

Másik irány

- Vagyis, ezek a tulajdonságok **szükségesek** az FO-definiálhatósághoz
- Belátjuk, hogy **elégségesek** is

Fa felbontása

- Elég nagy fában biztos van idempotens
  - Ha N állapot van, akkor N^N magas fa már elég nagy
- Így a nagy fák szétszedhetők blokkokra, amiket idempotensek kötnek össze
- Alkalmazva a három ekvivalenciát mint átírást, két „blokk-fa” átvihető egymásba, ha az olyan blokkjaik száma megegyezik, amikből egyiküknek kevesebb van, mint n
- Végeredmény LTT lesz $\rightarrow$ kész vagyunk

FO+TC

- FO és egyfajta tranzitív lezárás operátor
- Szintaxis ugyanaz, mint FO-é, plusz:

$F \rightarrow [TC_{\mathbf{x},\mathbf{y}}\ F(\mathbf{x},\mathbf{y})](\mathbf{s},\mathbf{t})$

Itt a négy változóvektor *k*-*k* darab szabad változóból áll; ennek szabad változói (**s**,**t**)

Szemantika:

akkor igaz, ha (**s**,**t**) benne van az F által megadott reláció tranzitív lezártjában

FO+TC példa

- FO(<)-ben:

  $\exists y.(x <_{\text{ch}}^{*} y \wedge p_a(y))$

- FO+TC-ben:

  $\exists y.([\text{TC}_{u,v}(u <_{ch} v)](x,y) \wedge p_a(y))$

Innen látszik, hogy FO(<) $\subseteq$ FO+TC

FO+TC vajon mi

- Az FO+TC-ben kifejezhető lekérdezések osztálya egybeesik az NL-ben kiszámítható lekérdezésekével
- Ha *determinisztikus* tranzitív lezárásunk van, arra

$$\text{FO+dTC} = \text{dPTW}$$

FO+TC

- FO+posTC: olyan FO+TC, amiben TC csak páros sok negáció hatáskörében áll

FO+TC = FO+posTC = PTW

Vége az első résznek

- Benedikt and Segoufin: *Regular tree languages definable in FO*
  LNCS vol. 3404, p. 327-339, 2005
- Geerts and Kuijpers: *Deciding termination of query evaluation in transitive-closure logics for constraint databases*
  LNCS vol. 2572, pp. 190-260, 2003.
- Libkin: *Logics for unranked trees – an overview*
  LNCS vol. 3580, p. 35-50, 2005.
- Neven: *Automata, logic, and XML*
  Invited talk at CSL 2002.
- Neven and Schwentick: *Expressive and efficient pattern languages for tree-structured data*
  PODS 2000: p. 145-156, 2000.
- Rabinovich: *Expressive power of temporal logics*
  LNCS vol 2421, p. 57-75, 2002.
- Walukiewicz: *Automata and logic*
- Walukiewicz: *Monadic second order logic on tree-like structures*
  TCS vol. 275, p. 311-346, 2002.