Bevezetés a szemantikus technológiákba

Szeredi Péter

szeredi@cs.bme.hu
BME Számítástudományi és Információelméleti Tanszék

Korábbi társelőadók: Lukácsy Gergely, Zombori Zsolt, Csorba János

2014 tavaszi félév

I. rész
Bevezetés

Szemantikus technológiák

- Wikipedia: Semantics ⇒ the study of meaning
- Szemantikus technológiák – (jelentést feldolgozó, azaz) tudásalapú technológiák
  - tudásbázis (Knowledge Base, KB) – formális, gép által kezelhető
  - kereső, információ-kinyerő algoritmusok a tudásbázison
- Korunk legnagyobb „tudásbázisa” a világháló
  - az információ-kinyerés főleg szövegegyezéses keresésen alapul
- A szemantikus világháló jelmondata:
  - A gép ne csak olvassa, értse is a weben található szövegeket!
- A szemantikus világháló eszközei:
  - Metainformáció társítás
  - Ontológiaépítés – háttértudás formalizálása
  - Automatikus következtetési módszerek
- A tudásábrázolás „ősi” eszköze a logika
- A szemantikus világháló eszköztára alkalmazható más környezetben is, pl. szemantikus integrációra

A kurzus felépítése

- Alapok: A világháló napjainkban (tankönyv 1. fejezet), logikai alapismeretek
- Ontológiák és leíró logikák (tankönyv 4.-6. fejezet)
  - Leíró logikák: $\mathcal{AL}$, $\mathcal{ALC}$, $\mathcal{SHIQ}$, $\mathcal{SROIQ}(\mathcal{D})$, ...
    - Háttértudás – ún. T-doboz (TBox, Terminology Box)
    - Metainformációk – ún. A-doboz (ABox, Assertion Box)
  - Következtetés leíró logikákon: főleg ún. tablóalgoritmusok
  - Egy egyszerű következtető megvalósítása Haskellben
- A szemantikus világháló informatikai oldala (a tankönyv 2, 3. és 7. fejezete)
  - RDF – metainformációk
  - RDFS – egyszerű háttértudás formalizálása
  - Az RDF használata
  - Ontológiák a Weben: OWL – Web Ontology Language
  - Protegé ontológiaépítő eszköz
- Tankönyv: Szeredi Péter, Lukácsy Gergely, Benkő Tamás: A szemantikus világháló elmélete és gyakorlata, Typotex, 2005

Tartalom

A világháló napjainkban

- A világháló tartalma
  - Heterogén szemantikájú és szintaktikájú dokumentumok
  - Eltérő típusok (szöveg, kép, hang, videó, . . .)
  - Eltérő formátumok (pdf, ps, word, txt,. . .)
  - Eltérő nyelvek (magyar, angol, Pascal, C, . . .)
  - Nem ellenőrzött (bárki bármit közzétehet)
- Keresés a világhálón
  - Oldalak begyűjtése (keresőrobotok)
  - Indexelés (tárgymutató készítése, fontos kifejezések kigyűjtése)
  - Kérdés értelmezése, keresés az indexben
  - Találatok sorrendezése és visszaadása

A világháló napjainkban (folyt.)

- Oldalak begyűjtése
  - Hosszadalmas (rengeteg adat)
  - Rendszeres frissités szükséges
  - Ha nincs link, nincs begyűjtés
- Indexelés
  - Dokumentum elemzése nehéz feladat
  - Mik a fontos kifejezések? Előbb meg kellene érteni . . .
  - Szavak gyakorisága jó heurisztika, de félrevezető lehet
  - Gépelési hibák, nem szabványos html
  - Eredménye egy jól karbantartott, tömör, strukturált, viszonylag kicsi adathalmaz

Keresés a világhálón

- Vektortér modell
  - Minden dokumentum és a kérdés egy-egy (irány)vektornak felel meg
  - Az irányvektorok által bezárt szög jellemzi a szövegek távolságát
  - Természetes nyelven megfogalmazott kérdésre jó
  - Kulcsszavas keresésre nem jó
- Boole modell
  - Csak azt figyeljük, hogy milyen kifejezések fordulnak elő az oldalon illetve a kérdésben
  - A hangsúly a keresés utáni rangsoroláson
  - Rangsoroláshoz különféle heurisztikák
    - Szavak gyakorisága, előfordulás helye (cím, bevezetés), fontméret, szín, korábbi felhasználók reakciói, . . .
    - A Google is valami ilyesmit használ

Keresés a világhálón (folyt.)

- Sorrendezés linkstruktura alapján
  - A fenti szempontok mind könnyen manipulálhatóak
  - Nehezen befolyásolható kritériumok kerülnek előtérbe
  - Link körüli horgony szöveg (anchor text) jelentősége: többet számít, hogy más mit mond rólunk, mint hogy mi mit mondunk magunkról
  - Érdekes oldalak által sokat hivatkozott oldal érdekesebb (csupán a linkstruktura alapján)
- Mérőszámok a keresés jellemzésére
  - Precizitás: releváns visszadott / visszaadott
  - Visszahívás: releváns visszaadott / releváns
  - Egymás ellen dolgoznak
  - Manapság tipikusan:
    - Kis precizitás (rengeteg érdektelen találat)
    - Nagy visszahívás (ritka, hogy a számunkra fontos oldalt ne találja meg a kereső)

Problémák a Webes kereséssel

- Hatalmas és változékony a világháló
- Mély Web
  - Lekérdezhető adatbázisban tárolt tartalom (Web nagyrésze!!!)
  - Nem szöveges tartalom
- Szemantika hiánya
  - Jelentés helyett szöveges alakkal dolgozunk
  - A keresés eredménye az információ tényleges reprezentációjától (és nem a tartalmától) függ
  - Nyelvi korlátok
  - Képekhez, hangokhoz semmilyen jelentést nem tudunk társítani
  - Nem tudunk következtetni (szinonimák, taxonómiák)

Egyszerű válaszok a Webes keresés problémáira

- Metakeresők: több kereső találatainak összevetésével adnak eredményeket
- Fókuszált keresők: kisebb méret, könnyebb frissíteni, jobb precizitás és visszahívás
- Szemantika megragadása
  - Kézi indexelés
    - Katalógust készítünk (YAHOO)
    - Ember szolgáltatja a szemantikát
    - Garantált minőség
    - Lassú
    - Melléktémák kimaradnak
    - Következtetés továbbra is hiányzik
  - Metainformáció elhelyezése a Weben
    - Metainformáció – olyan adat, amely az adott weblapról szól
    - Pl. link egy másik oldalra, szerző neve, dokumentum módosítási ideje
    - Jelenleg a metainformáció is heterogén formában van

A jövő: a Szemantikus Világháló

- Az oldalakhoz kapcsolódó metainformáció és a következtetéshez szükséges háttértudás egységes alakban jelenik meg (ontológia)
  - Péter gyereke Pál (metainformáció)
  - Szülő az, akinek van gyereke (háttértudás)
  - Következmény: Péter szülő
- Erőforrásainkhoz metaadatokat társítunk
- Mi lehet erőforrás? Bármi, ami egyedileg azonosítható (egy honlap, honlap része, kép videó, egy hardware eszköz, állomány)
- HTML-ben is van metaadat: <META> tag
  - Nagyon korlátozott, csak néhány attribútum
  - Csak a honlap egészéről szólhat
- A különféle formátumú adatforrásaink számára lehetővé tesszük, hogy metaadatot szolgáltassanak magukról
- A metaadatok így már egységesek, strukturáltak, gépi feldolgozásra (következtetésre) alkalmasak

Tartalom

Logikai alapok: az elsőrendű predikátumkalkulus

- A logika nézetei
  - Szintaxis (Mik a helyes mondatok?)
  - Szemantika — modellelmélet (Mit jelentenek a mondatok?)
  - Bizonyításelmélet (Hogyan juthatunk igaz mondatokhoz?)
  - Pragmatika (Mire jó az egész?)
- Szintaxis
  - Logikai és nem-logikai szimbólumok
  - Kifejezések (objektumok megnevezésére)
  - Formulák (állítások leírására)

Példa: az ontológiák iránt érdeklődő szakemberek klubja

Szintaxis:

- Lerögzítjük a fogalomrendszerünket („miről beszélünk”) azaz a szignatúrát: a függvény- és reláció**jelek** listáját, pl.
  - *klubtag/1* (egyargumentumú) reláció**jel**: $klubtag(x) \equiv x$ a klub tagja.
  - *resztVett/2* reláció**jel**: $resztVett(x, y) \equiv x$ részt vett az y előadáson.
  - *eloadas/1* reláció**jel**: $eloadas(y) \equiv y$ egy előadás.
  - *eloadoja/1* függvény**jel**: $eloadoja(y)$ az y előadás előadója.
  - *erdekli/2* reláció**jel**: $erdekli(x, z) \equiv x$-et érdekli a z tudományterület.
  - *ontologia/0* (nullargumentumú függvény**jel**, azaz konstans**jel**): $ontologia \equiv$ az ontológia tudományterülete.
- Állításokat (axiómákat) fogalmazunk meg
  - Az előadás előadója résztvevője is az előadásnak: $\forall y.(eloadas(y) \rightarrow resztVett(eloadoja(y), y))$
  - Aki részt vett egy előadáson, az klubtag és érdeklik az ontológiák: $\forall x.(\exists y.resztVett(x, y) \rightarrow (klubtag(x) \land erdekli(x, ontologia))$

Bizonyításelmélet:

- Állítások bizonyítása következtetési szabályokkal – szintaktikus következmény

Példa: az ontológiaklub, folyt.

Szemantika:

- Lerögzítünk egy „világot” (interpretációt, modellt):
  - alaphalmaz: az klubelőadások eddigi résztvevői, előadásai, érdeklődési területek stb.
  - megadjuk az egyes reláció- és függvényjelek értelmezését: a *klubtag*ok halmazát, a *resztVett*-párok halmazát stb.
- A szemantika leírja az adott világban a formulák jelentését, pl.:
  - meghatározható az alábbi állítás igazságértéke: $\forall y.(eloadas(y) \rightarrow \exists x_1, x_2.(resztVett(x_1, y) \land resztVett(x_2, y) \land x_1 \neq x_2)))$
- Általánosíthatunk: felállítjuk az X-klubok elméletét, axiómarendszerét, pl.
  - A klubnak legalább egy előadása, és minden előadásnak legalább két résztvevője van.
  - Minden résztvevő klubtag és érdeklődik a klub alaptémája iránt. ...
- Megkereshetjük az axiómarendszerünk *következmény*-állításait:
  - Egy $\mathcal{A}$ állításhalmaz következménye egy S állítás, ha S igaz minden olyan világban, ahol $\mathcal{A}$ minden állítása igaz (szemantikus követk.).
  - Pl. a fenti axiómákból következik, hogy az alaptéma iránt legalább ketten érdeklődnek.

Az elsőrendű predikátumkalkulus szintaxisa – szimbólumok

- logikai szimbólumok
  - központozási jelek: (,) .
  - logikai összekötő jelek:
    - $\neg$ (negáció — „nem”),
    - $\wedge$ (konjunkció — „és”),
    - $\vee$ (unió — „vagy”),
    - $\exists$ (egzisztenciális kvantor — „létezik olyan . . . ”),
    - $\forall$ (univerzális kvantor — „minden . . . -ra igaz, hogy . . . ”),
    - $=$ (egyenlőség)
  - változók: $x, y, z, x_1, y_1, \ldots, x_i, \ldots$
- nem-logikai szimbólumok
  - függvényjelek: a, b, c (konstansok azaz 0-argumentumú függvényjelek), $f, g, h, \ldots$
  - predikátumjelek: $p, q, r, \ldots$
  - mind a függvény-, mind a predikátumjeleknek van egy rögzített ≥ 0 argumentumszáma (aritása)
  - szignatúra (vagy nyelvtípus): a használni kívánt függvény- és predikátumjelek felsorolása (aritással együtt)

Az elsőrendű predikátumkalkulus szintaxisa — folyt. (1)

- Kifejezések (*terms*) — olyan jelsorozatok, amelyek a világ egy objektumát nevezik meg.
  - Minden változójel kifejezés.
  - Ha $t_1, \ldots, t_n$ kifejezések és f egy n-argumentumú függvényjel, akkor $f(t_1, \ldots, t_n)$ is egy kifejezés.
    (Speciálisan: $b()$ is egy kifejezés, ahol b tetszőleges konstansjel.)
  - Az elsőrendű logika kifejezései: a fenti két feltételt kielégítő legszűkebb halmaz.
- Formulák (*formulae*) — egy állítást megfogalmazó jelsorozatok.
  - Ha $t_1, \ldots, t_n$ kifejezések és p egy n-argumentumú predikátumjel, akkor $p(t_1, \ldots, t_n)$ is egy állítás (*elemi* vagy *atomi* állítás, ill. *atom*).
  - Ha t_1 és t_2 kifejezések, akkor $t_1 = t_2$ egy formula (ez is *elemi* állítás).
  - Ha α és β formulák, x egy változójel, akkor $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\exists x.\alpha)$, $(\forall x.\beta)$ szintén formulák.
  - Az elsőrendű logika formulái (*well-formed-formulas, wff*): a fenti rekurzív definíciót kielégítő legszűkebb halmaz.

Az elsőrendű predikátumkalkulus szintaxisa — folyt. (2)

- Szintaktikus édesítőszerek: zárójelek, pontok elhagyása, beszúrása, 0-argumentumú függvény- ill. predikátumjelek utáni () elhagyása, $\neq$ bevezetése stb.
- Rövidítések — további édesítőszerek
  - $(\alpha \rightarrow \beta)$ jelentése: $(\neg\alpha \vee \beta)$
  - $(\alpha \equiv \beta)$ jelentése: $((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$
- Kvantorok hatásköre
  - Kötött (*bound*) változó: olyan változó-előfordulás, amely egy kvantor hatáskörében van. Pl. x minden előfordulása kötött egy $\exists x.\alpha$ vagy egy $\forall x.\alpha$ részformula belsejében.
  - Szabad (*free*) változó: olyan változó-előfordulás, amely nincs egy kvantor hatáskörében.
- Mondat (*sentence*): olyan formula, amelyben nincs szabad változó (szokás zárt formulának is nevezni)

Az elsőrendű predikátumkalkulus szemantikája

- A szintaxis megadja azon jelsorozatokat, amelyek helyes formulák
- Alfred Tarski modellelméleti megközelítése szerint a szemantika megadja egy tetszőleges helyes formula jelentését (durván igazságértékét), *feltéve, hogy lerögzítjük a függvények és predikátumok jelentését*, azaz megadunk egy interpretációt
- Interpretáció: $\mathcal{I} =< \Delta, I >$
  - Δ egy tetszőleges alaphalmaz (*domain*)
  - I egy (felső indexként jelölt) hozzárendelés, amely minden
    - n-argumentumú f függvényjelhez egy Δ-n értelmezett n-argumentumú függvényt rendel: $f^I \in \Delta \times \ldots \times \Delta \mapsto \Delta$
      (f^I az f függvény**jel**hez rendelt függvény)
    - n-argumentumú p predikátumjelhez egy Δ-n értelmezett n-argumentumú relációt rendel: $p^I \subseteq \Delta \times \ldots \times \Delta$
      (p^I a p predikátum**jel**hez rendelt reláció)
- Megjegyzés: Az, hogy az f^I függvény ill. az p^I reláció „kiszámítása” hogyan írható le, nem tartozik a logika területére!

Az elsőrendű predikátumkalkulus szemantikája, folyt.

- Egy interpretáció kontextusában minden változómentes kifejezéshez az alaphalmaz egy elemét rendelhetjük hozzá. Hasonlóan minden zárt formulához igazságértéket rendelhetünk.
- Változót is tartalmazó kifejezés ill. szabad változót tartalmazó formula kiértékeléséhez szükség van egy ún. változó-értékelésre (*valuation*):
  - A változó-értékelés egy φ függvény, amely minden változójelhez az alaphalmaz egy elemét rendeli: $\varphi(x) \in \Delta$
  - Jelölés: $\varphi[x \mapsto d]$ az az értékelés, amely minden x-től különböző változóhoz ugyanazt az értéket rendeli mint φ, míg x-hez a $d \in \Delta$-t.
- Adott $\mathcal{I} =< \Delta, I >$ interpretáció és φ értékelés mellett rekurzívan definiáljuk egy tetszőleges t kifejezés $t^{\varphi,I}$ jelentését:
  - Ha x egy változó, akkor $x^{\varphi,I} = \varphi(x)$,
  - Ha $t_1, \ldots, t_n$ kifejezések és f egy n-argumentumú függvényjel, akkor $f(t_1, \ldots, t_n)^{\varphi,I} = f^I(t_1^{\varphi,I}, \ldots, t_n^{\varphi,I})$

Az elsőrendű predikátumkalkulus szemantikája, folyt. (2)

- Adott $\mathcal{I} =< \Delta, I >$ interpretáció és φ értékelés mellett rekurzívan definiáljuk a formulák igazságértékét:
  $\mathcal{I} \models_\varphi \alpha$: az $\mathcal{I}$ interpretáció a φ értékelés mellett kielégíti az α formulát.
  - $\mathcal{I} \models_\varphi p(t_1, \ldots, t_n) \Leftrightarrow \langle d_1, \ldots, d_n \rangle \in \mathcal{P}$, ahol $\mathcal{P} = p^I$ és $d_i = t_i^{\varphi,I}$.
  - $\mathcal{I} \models_\varphi t_1 = t_2 \Leftrightarrow d_1$ és d_2 Δ ugyanazon eleme, ahol $d_i = t_i^{\varphi,I}$.
  - $\mathcal{I} \models_\varphi \neg\alpha \Leftrightarrow$ nem teljesül az, hogy $\mathcal{I} \models_\varphi \alpha$.
  - $\mathcal{I} \models_\varphi \alpha \wedge \beta \Leftrightarrow$ teljesül $\mathcal{I} \models_\varphi \alpha$ és teljesül $\mathcal{I} \models_\varphi \beta$.
  - $\mathcal{I} \models_\varphi \alpha \vee \beta \Leftrightarrow \mathcal{I} \models_\varphi \alpha$ és $\mathcal{I} \models_\varphi \beta$ közül legalább az egyik teljesül.
  - $\mathcal{I} \models_\varphi \forall x.\alpha \Leftrightarrow$ minden $d \in \Delta$ elemre igaz, hogy $\mathcal{I} \models_{\varphi[x \mapsto d]} \alpha$.
  - $\mathcal{I} \models_\varphi \exists x.\alpha \Leftrightarrow$ van olyan $d \in \Delta$, hogy $\mathcal{I} \models_{\varphi[x \mapsto d]} \alpha$.
- Belátható, hogy zárt formula (mondat) esetén a kielégítés nem függ a változó-értékeléstől, ilyenkor az $\mathcal{I} \models \alpha$ alakot használjuk, és azt mondjuk, hogy α **igaz** az $\mathcal{I}$ interpretációban.
- Jelölések (S mondathalmaz, α mondat):
  - $\mathcal{I} \models S$ ($\mathcal{I}$ az S **modellje**): S minden eleme igaz $\mathcal{I}$-ben.
  - $S \models \alpha$ (S-nek logikai vagy szemantikus következménye α): bármely $\mathcal{I}$ interpretáció esetén, ha $\mathcal{I} \models S$ akkor $\mathcal{I} \models \alpha$ is fennáll.

Bizonyítások az elsőrendű predikátumkalkulusban

- Bizonyításelmélet: a matematika formalizálja önmagát.
- Következtetési rendszer: következtetési szabályok halmaza.
- Következtetési szabály:
  - olyan (szintaktikus) transzformáció, amely
  - egy vagy több mondatból egy új mondatot állít elő.
- Szintaktikus következményfogalom: $S \vdash \alpha$ (S-ból levezethető α) akkor és csak akkor ha:
  - létezik mondatok olyan $\alpha_1, \ldots, \alpha_n$ sorozata,
  - ahol minden i-re
    - vagy $\alpha_i \in S$;
    - vagy α_i az őt megelőző mondatokból egy következtetési szabállyal áll elő.
  - Egy következtetési rendszer **helyes** (*sound*), ha $S \vdash \alpha \Rightarrow S \models \alpha$ (amit kiköveztet, az igaz).
  - Egy következtetési rendszer **teljes** (*complete*), ha $S \models \alpha \Rightarrow S \vdash \alpha$ (ami igaz, azt kiköverteti).

Teljesség és eldönthetőség

- Kurt Gödel teljességi tételében megmutatta, hogy az elsőrendű logika egy adott $\vdash$ következtetési rendszere teljes (és triviálisan helyes is), tehát $S \models \alpha \Leftrightarrow S \vdash \alpha$ igaz.
  - Zárójeles megjegyzés: a teljességi tétel rövid alakja megjelenik az ALP logójában (lásd pl: http://www.cs.nmsu.edu/ALP/)

    Association for Logic Programming
- Gödel teljességi tételéhez kapcsolódva az is megmutatható, hogy egy adott elsőrendű axiómarendszer következményei rekurzíve felsorolhatóak, azaz írható olyan program, amely minden következményt előbb-utóbb felsorol.
- Az elsőrendű logika nem eldönthető: nem létezhet olyan program, amely egy véges axiómarendszer esetén egy tetszőleges mondatról eldönti, hogy az következmény-e, vagy sem, egy csak a mondattól függő véges időn belül.
- Az elsőrendű logikának vannak eldönthető résznyelvei, a legtöbb Leíró Logika ilyen résznyelvvel ekvivalens.

II. rész

Leíró Logikák

1 Bevezetés

2 Leíró Logikák

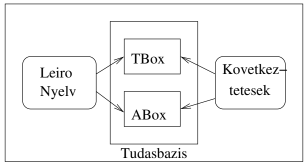
3 A szemantikus világháló

Tartalom

Leíró logika mint az elsőrendű predikátumkalkulus résznyelve

- Szignatúra:
  - szerepnév (role name, R) – kétargumentumú relációjel
  - fogalomnév (concept name, A) – egyargumentumú relációjel
  - egyednév (individual name, $a, \ldots$) – konstansjel, azaz 0-arg. fvjel
  - nem konstans függvényjelek, kettőnél nagyobb aritású relációjelek, változók nincsenek
- Fogalomkifejezések, pl. $\exists$szülője.Opt – azon egyedek halmaza, akiknek van optimista szülője
- Terminológiai axiómák (T-doboz, TBox):
  - Egy egyszerű ($\mathcal{ALE}$ nyelvű) állítás pl: $\exists$szülője.Opt $\sqsubseteq$ Opt
  - Azok halmaza akiknek van optimista szülője *része-egyenlő* az optimisták halmazának, azaz akinek van opt. szülője, az maga is opt.
  - Elsőrendű megfelelője: $\forall x.(\exists y.(sz(x,y) \wedge Opt(y)) \rightarrow Opt(x))$
- Adataxiómák (A-doboz, ABox):
  - Példa: Opt(JÁKOB), szülője(JÓZSEF, JÁKOB)
  - A fenti axiómákból, kikövetkeztethető, hogy Opt(JÓZSEF),
- A leíró logikák zöme eldönthető: van algoritmus állítások igazságának eldöntésére

Leíró logikák mint a tudásreprezentáció eszközei



- Tudásbázis (KB, knowledge base) = T-doboz (TBox) + A-doboz (ABox):
- T-doboz = terminológiai doboz = terminológiai axiómák halmaza: fogalmakról (és szerepekről) szóló állítások (anya az aki nőnemű és van gyereke)
- A-doboz = adatdoboz = adataxiómák halmaza: tudásunk az objektumokról (Éva anya)
- Következtetések: T-doboz: egy fogalomleírás kielégíthető (értelmes), annak megállapítása hogy az egyik fogalom egy másik általánosítása (fogalom-hierarchia),
  A-doboz: eldöntendő, hogy egy egyed egy fogalom példánya-e, egy fogalomkifejezést kielégítő objektumok meghatározása

Példák terminológiai axiómákra

- Az Anya nem más, mint olyan Ember aki Nőnemű és van gyereke.
  $$\text{Anya} \equiv \text{Ember} \sqcap \text{Nőnemű} \sqcap \exists \text{gyereke}.\top$$
- Minden Tigris Emlős.
  $$\text{Tigris} \sqsubseteq \text{Emlős}$$
- A boldog emberek gyerekei is boldogak.
  $$\text{Boldog} \sqcap \text{Ember} \sqsubseteq \forall \text{gyereke}.\text{Boldog}$$
- A gyermektelen emberek boldogak
  $$\forall \text{gyermeke}.\bot \sqcap \text{Ember} \sqsubseteq \text{Boldog}$$
- A gyereke viszonyban levők egyben leszármazottja viszonyban is vannak.
  $$\text{gyereke} \sqsubseteq \text{leszármazottja}$$
- A szülője kapcsolat a gyereke kapcsolat megfordítottja (inverze).
  $$\text{szülője} \equiv \text{gyereke}^{-}$$
- A leszármazottja reláció tranzitív
  $$\text{Trans(leszármazottja)}$$

Tartalom

Az $\mathcal{AL}$ nyelv szintaxisa

- Az $\mathcal{AL}$ (Attributive Language) fogalomkifejezések (röviden fogalmak):

$C \rightarrow$	$A \mid$	(atomi fogalom)	egy halmaz, pl: Ember
	$\top \mid$	(tetőjel, top)	az összes objektum halmaza
	$\bot \mid$	(fenékjel, bottom)	az üres halmaz
	$\neg A \mid$	(atomi negálás)	
	$C \sqcap D \mid$	(metszet)	
	$\forall R.C \mid$	(értékkorlátozás)	azok, akik minden R-je C-beli
	$\exists R.\top$	(egysz. létezési korl.)	azok, akiknek létezik R-je

  A atomi fogalom (vagyis fogalomnév), C, D tetszőleges fogalmak
- Az $\mathcal{AL}$ nyelvben megengedett axiómák szintaxisa:
  $C \sqsubseteq D$ és $C \equiv D$
- Példák fogalomkifejezésekre:
  Ember $\sqcap \neg$Nőnemű
  Ember $\sqcap \forall$gyereke.Nőnemű
  Ember $\sqcap \exists$gyereke.$\top$
- Példa axiómára: Kékszemű $\sqsubseteq \forall$szülője.Kékszemű

Az $\mathcal{AL}$ nyelv szemantikája

- Interpretáció: $\mathcal{I} =< \Delta, I >$
  - Δ az objektumok halmaza (nem lehet üres!).
  - Az I függvény atomi fogalmakhoz Δ részhalmazait, atomi szerepekhez Δ-n értelmezett 2-arg. relációkat rendel.
  - Az I hozzárendelés az alábbi módon kiterjeszthető tetsz. fogalmakra:

$$\begin{aligned}
\top^I &= \Delta \\
\bot^I &= \emptyset \\
(\neg A)^I &= \Delta \setminus A^I \\
(C \sqcap D)^I &= C^I \cap D^I \\
(\forall R.C)^I &= \{a \in \Delta \mid \forall b.(\langle a, b \rangle \in R^I \rightarrow b \in C^I)\} \\
(\exists R.\top)^I &= \{a \in \Delta \mid \exists b.\langle a, b \rangle \in R^I\}
\end{aligned}$$

- Az interpretációs jelölés egyszerűsítése: ha adott $\mathcal{I}$ ahol $\mathcal{I} =< \Delta, I >$, akkor a Δ alaphalmaz helyett $\Delta^{\mathcal{I}}$-t, C^I helyett $C^{\mathcal{I}}$-t, R^I helyett $R^{\mathcal{I}}$-t írunk.
- Axiómák igazságtartalma: $\mathcal{I} \models C \sqsubseteq D$ csakkor ha $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
  $\mathcal{I} \models C \equiv D$ csakkor ha $C^{\mathcal{I}} = D^{\mathcal{I}}$

Az $\mathcal{AL}$ nyelvcsalád: az $\mathcal{U}$, $\mathcal{E}$, $\mathcal{N}$, $\mathcal{C}$ nyelvkiterjesztések

- Unió: $C \sqcup D$
$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \qquad (\mathcal{U})$$
- Teljes létezési korlátozás: $\exists R.C$
$$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \exists b. \langle a, b \rangle \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \qquad (\mathcal{E})$$
- Számosság-korlátozások (nem-minősítettek): $(\geqslant n\, R)$ és $(\leqslant n\, R)$
$$(\geqslant nR)^{\mathcal{I}} = \{\, a \in \Delta \mid \quad |\{b \mid \langle a, b \rangle \in R^{\mathcal{I}}\}| \geq n \,\} \qquad (\mathcal{N})$$
$$(\leqslant nR)^{\mathcal{I}} = \{\, a \in \Delta \mid \quad |\{b \mid \langle a, b \rangle \in R^{\mathcal{I}}\}| \leq n \,\}$$
Figyelem: $(\geqslant nR.C)$ (például az hogy valakinek van legalább 3 kékszemű gyereke) már minősített korlátozás, ezt az $\mathcal{N}$ nyelvkiterjesztésen túlmutat.
- (Teljes) negálás, azaz komplemens-képzés: $\neg C$
$$(\neg C)^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}} \qquad (\mathcal{C})$$
- pl: Ember $\sqcap$ ($\leqslant$ 1gyereke $\sqcup$ ($\geqslant$ 3gyereke $\sqcap$ $\exists$gyereke.Nőnemű)).

Az $\mathcal{AL}$ nyelvcsalád – különböző erejű nyelvek

- A $\mathcal{AL}$ nyelvet az előbbi konstruktorok egy halmazával kiegészíthetjük:
$$\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}]$$
ez összesen $2^4 = 16$ nyelvet jelent.
- Mivel $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$ és $\exists R.C \equiv \neg\forall R.\neg C$, az unió és a teljes létezési korlátozás kifejezhető a (teljes) negálás segítségével. Tehát minden $\mathcal{ALUE}$ kifejezéshez van vele ekvivalens $\mathcal{ALC}$ kifejezés.
- Visszafelé, minden $\mathcal{ALC}$ kifejezéshez előállítható vele ekvivalens $\mathcal{ALUE}$ kifejezés, úgy hogy a negációt kiküszöböljük, ill. „bevisszük” az atomi fogalmak elé, a $\neg\neg C \equiv C, \neg\top \equiv \bot, \neg\bot \equiv \top, \neg(C \sqcap D) \equiv \neg C \sqcup \neg D, \neg\exists R.\top \equiv \forall R.\bot, \neg\forall R.C \equiv \exists R.\neg C$ azonosságok ismételt alkalmazásával.
- Tehát $\mathcal{ALUE}$ és $\mathcal{ALC}$ azonos kifejező erejű, és így ($\mathcal{ALC} = \mathcal{ALCU} = \mathcal{ALCE} = \mathcal{ALCUE} = \mathcal{ALUE}$).
- Ha a $\mathcal{C}$ nyelvkiterjesztést elhagyjuk, a maradék 8 nyelvről belátható, hogy ezek páronként különbözőek.

A leíró nyelvek és az elsőrendű logika

- A fogalmak átírhatók elsőrendű logikai kifejezésekké
- Az átírás minden C fogalomkifejezésnek egy $\Phi_C(x)$ formulát feleltet meg.
- Az atomi fogalmak(A) és szerepek(R) unáris illetve bináris predikátumok lesznek ($A(x), R(x, y)$).
- A metszetet, az uniót, a negálást egyszerűen a logikai megfelelőjére írjuk át.
- A különféle korlátozások a következő módon íródnak át:

$$\begin{aligned}
\Phi_{\exists R.C}(y) &= \exists x.\,(R(y,x) \wedge \Phi_C(x)) \\
\Phi_{\forall R.C}(y) &= \forall x.\,(R(y,x) \rightarrow \Phi_C(x)) \\
\Phi_{(\geqslant n\,R)}(x) &= \exists y_1, \ldots, y_n. \left(R(x,y_1) \wedge \cdots \wedge R(x,y_n) \wedge \bigwedge_{i<j} y_i \neq y_j \right) \\
\Phi_{(\leqslant n\,R)}(x) &= \forall y_1, \ldots, y_{n+1}. \left(R(x,y_1) \wedge \cdots \wedge R(x,y_{n+1}) \rightarrow \bigvee_{i<j} y_i = y_j \right)
\end{aligned}$$

Tartalom

A $\mathcal{SHIQ}$ leíró logikai nyelv áttekintése

- A $\mathcal{SHIQ}$ rövidítés jelentése
  - $\mathcal{S} \equiv \mathcal{ALC}_{\mathcal{R}^+}$ (az $\mathcal{ALC}$ nyelv kiegészítve tranzitív szerepekkel), azaz egyes szerepekről (pl. őse) kijelenthetjük, hogy tranzitívak.
  - $\mathcal{H} \equiv$ szerephierarchiák. Egy szerephierarchia $R \sqsubseteq S$ alakú állítások halmaza, pl. minden barátja kapcsolat egyben ismerőse kapcsolat is: barátja $\sqsubseteq$ ismerőse.
  - $\mathcal{I} \equiv$ inverz szerepek: egy R szerep mellett annak R^- inverzét is használhatjuk, pl. gyereke$^-$ $\equiv$ szülője.
  - $\mathcal{Q} \equiv$ minősített számosság-korlátozások, azaz $(\leqslant n\, R.C)$ és $(\geqslant n\, R.C)$ alakú fogalomkifejezések (az $\mathcal{N}$ általánosítása) pl. azon emberek akiknek legalább 3 okos gyereke van: $(\geqslant 3\,\text{gyereke.Okos})$
- A $\mathcal{Q}$ minősített számosság-korlátozásokat két lépésben vezetjük be:
  - $\mathcal{F} \equiv$ funkcionális korlátozások
    - $(\leqslant 1\, R)$ és (ennek negáltjaként) $(\geqslant 2\, R)$ alakú fogalomkifejezések
  - általános minősített számosság-korlátozások

Miért pont a $\mathcal{SHIQ}$ nyelv?

- A tranzitív szerepek és a szerephierarchiák fontosak a rész-egész kapcsolatokban, az (OO) öröklődési kapcsolatokban
  - Szerepnevek és inverzeik
    - $\text{része}(\text{Autó}, \text{Henger}) \equiv$ Az Autónak része a Henger.
    - $\text{tartalmazója}(\text{Henger}, \text{Autó}) \equiv$ A Hengernek tartalmazója az Autó.
    - „-nek/nak” csak a baloldalon lehet!!!
  - Rész-egész kapcsolatok és inverzeik elnevezése
    - (közvetlen) komponense (hasComponent) – befoglalója (iscomponentOf)
    - része (hasPart) – tartalmazója (isPartOf) (az előző bajuszbeli szerepek tranzitív lezárása)
    - példák: $\text{kompononense}(\text{Autó}, \text{Motor})$, $\text{kompononense}(\text{Motor}, \text{Henger}), \ldots$
    - ugyanez megfordítva: $\text{befoglalója}(\text{Motor}, \text{Autó})$, $\text{befoglalója}(\text{Henger}, \text{Motor}), \ldots$
    - következmény: $\text{része}(\text{Autó}, \text{Henger}) \equiv \text{tartalmazója}(\text{Henger}, \text{Autó})$

Miért pont a $\mathcal{SHIQ}$ nyelv? (2)

- Példa: nukleáris reaktorok fogalmi rendszere
  - Axiómák:

$$\begin{array}{lll}
\text{befoglalója} \sqsubseteq \text{tartalmazója} & & (\text{isComponentOf} \sqsubseteq \text{isPartOf}) \\
\text{Vezérrúd} \sqsubseteq \text{Eszköz} \sqcap \exists\, \text{befoglalója}.\text{Reaktormag} & & \\
\text{Reaktormag} \sqsubseteq \text{Eszköz} \sqcap \exists\, \text{befoglalója}.\text{Reaktor} & & \\
\text{Trans}(\text{tartalmazója}) \equiv \text{tartalmazója egy tranzitív reláció} & &
\end{array}$$

  - A példában kiköveztethető, hogy

$$\text{Vezérrúd} \sqsubseteq \exists\, \text{tartalmazója}.\text{Reaktor}$$

- Az inverz szerepek lehetővé teszik, hogy a rész-egész kapcsolatokat mindkét irányban leírjuk, pl. a tartalmazója (is_part_of) mellett használhatjuk a része (has_part) szerepeket.
  - Például definiálhatjuk a VeszélyesReaktor fogalmat így:

$$\text{Reaktor} \sqcap \exists\, \text{része}.\text{Hibás} \sqsubseteq \text{VeszélyesReaktor}$$

  - Ezután kiköveztethető, hogy

$$\text{Vezérrúd} \sqcap \text{Hibás} \sqsubseteq \exists\, \text{tartalmazója}.\text{VeszélyesReaktor}$$

Miért pont a $\mathcal{SHIQ}$ nyelv? (3)

- A számosság-korlátozások jelentősége:
  - A funkcionális korlátozások fontosak az Entity-Relationship fajtájú modellezésben leggyakrabban előforduló 0..1 multiplicitások leírására. Példa:

$$\text{Reaktor} \sqsubseteq \exists\, \text{vezérlője}.\text{Vezérlőegység} \sqcap (\leqslant\ 1\ \text{vezérlője})$$

    azaz: minden reaktornak pontosan egy vezérlője van, ami egy vezérlőegység.
  - A minősített számosság-korlátozásokkal az általános $n..m$ multiplicitások is leírhatók.
- A tranzitívitás és szerephierarchia jelentősége: az $\mathcal{SH}$ nyelvre és bővítményeire alkalmazható az ún. belsősítés (internalization), a fogalmi axiómák kiküszöbölésére.

Miért pont a $\mathcal{SHIQ}$ nyelv? (4)

- A inverz és a funkcionális korl. együttes bevezetésének jelentősége:
  - Az egyszerűbb leíró logikai nyelvek rendelkeznek az ún. véges modell tulajdonsággal:
    - Ha egy C fogalom kielégíthető egy $\mathcal{T}$ T-doboz felett – azaz van olyan modell, amelyben $\mathcal{T}$ axiómái fennálnak, és C nem üres – akkor van véges ilyen modell is.
  - Az $\mathcal{ALIF}$ és bővebb (pl. a $\mathcal{SHIQ}$) nyelvekre nincs véges modell tulajdonság:
    - $\mathcal{T}$ = $\{\top \sqsubseteq \exists \text{gy}.\top, \top \sqsubseteq (\leqslant 1\ \text{gy}^-)\}$: mindenkinek van gyereke, és legf. egy szülője.
    - $C = \forall \text{gy}^-.\bot$ (nincs szülője) kielégíthető $\mathcal{T}$ felett, de csak végtelen modellel.

Az egyes $\mathcal{SHIQ}$ nyelvkiterjesztések

- $\mathcal{S} \equiv \mathcal{ALC}_{\mathcal{R}^+}$, azaz $\mathcal{ALC}$ kiterjesztve tranzitív relációkkal
  - Egy elvetett lehetőség, a $^+$ szerepművelet bevezetése:
    - R^+ = az R tranzitív lezártja, a *legszükebb* olyan tranz. szerep, amely bővebb mint R
    - ez túl erős, túlmutat az elsőrendű logikán
  - Az $\mathcal{S}$ nyelv:
    - Az $\mathcal{ALC}$ nyelv fogalomkifejezései és axiómái
    - Trans(R) alakú axiómák: jelentésük: az R szerep tranzitív.
- $\mathcal{H}$ – szerephierarchia
  - $R \sqsubseteq S$ és $R \equiv S$ alakú szerepaxiómák (az '$\equiv$' kiküszöbölhető)
  - Az $\mathcal{SH}$ nyelvben definiálható egy „gyenge” tranzitív lezárási művelet:
    - Példa:

$$\text{Trans}(\text{leszármazottja})$$
$$\text{gyereke} \sqsubseteq \text{leszármazottja}$$

    - leszármazottja egy olyan tranzitív szerep, amely a gyereke szerepnél bővebb: leszármazottja $\sqsupseteq$ gyereke$^+$ (de nem feltétlenül a legszűkebb ilyen).

Az egyes $\mathcal{SHIQ}$ nyelvkiterjesztések (2)

- $\mathcal{I}$ – inverz szerepek
  - Első szerepkonstruktorunk a $^-$: R^- – az R szerep inverze
  - Pl: A gyereke$^-$ $\equiv$ szülője szerepaxióma és az alábbi fogalmi ax.-ák:

$$\begin{aligned} \text{JóSzülő} &\equiv \exists \text{gyereke}.\top \sqcap \forall \text{gyereke}.\text{Boldog} \\ \text{VidámGyermek} &\equiv \exists \text{szülője}.\text{JóSzülő} \end{aligned}$$

    következménye az alábbi fogalmi állítás: VidámGyermek $\sqsubseteq$ Boldog
  - A többszörös invertálás redundáns: $(R^-)^- \equiv R, ((R^-)^-)^- \equiv R^-$ stb.
  - Hasznos jelölés $\text{Inv}(R) = \begin{cases} S & \text{ha } R = S^- \text{ alakú} \\ R^- & \text{egyébként} \end{cases}$
- $\mathcal{F}$ – funkcionális korl.: $(\leqslant 1\ R)$ ill. negáltja $(\geqslant 2\ R)$ ($\mathcal{N}$ spec. esete)
- $\mathcal{Q}$ – minősített számosság-korlátozások ($\mathcal{N}$ általánosítása)
  - új fogalomkonstruktorok: $(\leqslant n\, R.C)$ ill. $(\geqslant n\, R.C)$ – azon egyedek halmaza, amikhez legfeljebb (ill. legalább) n, vele R kapcsolatban álló olyan egyed van, amely C-beli.
- Funkcionális és számosság-korlátozásban csak *egyszerű* szerep megengedett: olyan, amelynek nincs tranzitív része

$\mathcal{SHIQ}$ szintaxis: összefoglalás

- Fogalomkifejezések szintaxisa

$C \rightarrow$		A	*atomi fogalom*	$(\mathcal{AL})$
	$\mid$	$\top$	*tetőjel – univerzális fogalom*	$(\mathcal{AL})$
	$\mid$	$\bot$	*fenékjel – semmis fogalom*	$(\mathcal{AL})$
	$\mid$	$\neg C$	*negálás*	$(\mathcal{C})$
	$\mid$	$C_1 \sqcap C_2$	*metszet*	$(\mathcal{AL})$
	$\mid$	$C_1 \sqcup C_2$	*unió*	$(\mathcal{U})$
	$\mid$	$\forall R.C$	*érték-korlátozás*	$(\mathcal{AL})$
	$\mid$	$\exists R.C$	*létezési korlátozás*	$(\mathcal{E})$
	$\mid$	$(\geqslant n\, R_S.C)$	*minősített számosság-korlátozás*	$(\mathcal{Q})$
			(R_S: *egyszerű szerep, nincs tranzitív része*)	
	$\mid$	$(\leqslant n\, R_S.C)$	*minősített számosság-korlátozás*	$(\mathcal{Q})$

$\mathcal{SHIQ}$ szintaxis: összefoglalás (2)

- Szerepkifejezések szintaxisa

$$\begin{array}{llll} R \rightarrow & R_A & \textit{atomi szerep} & (\mathcal{AL}) \\ | & R^- & \textit{inverz szerep} & (\mathcal{I}) \end{array}$$

- Terminológiai állítások (axiómák) szintaxisa

$$\begin{array}{llll} T \rightarrow & C_1 \equiv C_2 & \textit{fogalomegyenlőségi axióma} & (\mathcal{AL}) \\ | & C_1 \sqsubseteq C_2 & \textit{fogalomtartalmazási axióma} & (\mathcal{AL}) \\ | & R_1 \equiv R_2 & \textit{szerepegyenlőségi axióma} & (\mathcal{H}) \\ | & R_1 \sqsubseteq R_2 & \textit{szereptartalmazási axióma} & (\mathcal{H}) \\ | & \mathrm{Trans}(R) & \textit{tranzitívszerep-axióma} & (\mathcal{R}^+) \end{array}$$

$\mathcal{SHIQ}$ szemantika

- Fogalomkifejezések szemantikája

$$\begin{array}{rcl} \top^\mathcal{I} & = & \Delta^\mathcal{I} \\ \bot^\mathcal{I} & = & \emptyset \\ (\neg C)^\mathcal{I} & = & \Delta^\mathcal{I} \setminus C^\mathcal{I} \\ (C_1 \sqcap C_2)^\mathcal{I} & = & C_1^\mathcal{I} \cap C_2^\mathcal{I} \\ (C_1 \sqcup C_2)^\mathcal{I} & = & C_1^\mathcal{I} \cup C_2^\mathcal{I} \\ (\forall R.C)^\mathcal{I} & = & \{a \in \Delta^\mathcal{I} \mid \forall b.\langle a,b\rangle \in R^\mathcal{I} \rightarrow b \in C^\mathcal{I}\} \\ (\exists R.C)^\mathcal{I} & = & \{a \in \Delta^\mathcal{I} \mid \exists b.\langle a,b\rangle \in R^\mathcal{I} \wedge b \in C^\mathcal{I}\} \\ (\geqslant n\, R.C)^\mathcal{I} & = & \{a \in \Delta^\mathcal{I} \mid |\{b \mid \langle a,b\rangle \in R^\mathcal{I} \wedge b \in C^\mathcal{I}\}| \geq n\} \\ (\leqslant n\, R.C)^\mathcal{I} & = & \{a \in \Delta^\mathcal{I} \mid |\{b \mid \langle a,b\rangle \in R^\mathcal{I} \wedge b \in C^\mathcal{I}\}| \leq n\} \end{array}$$

- Szerepkifejezések szemantikája

$$(R^-)^\mathcal{I} = \{\langle b,a\rangle \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid \langle a,b\rangle \in R^\mathcal{I}\}$$

$\mathcal{SHIQ}$ szemantika (2)

- Terminológiai axiómák szemantikája

$$\begin{array}{lcl} \mathcal{I} \models C_1 \equiv C_2 & \Leftrightarrow & C_1^\mathcal{I} = C_2^\mathcal{I} \\ \mathcal{I} \models C_1 \sqsubseteq C_2 & \Leftrightarrow & C_1^\mathcal{I} \subseteq C_2^\mathcal{I} \\ \mathcal{I} \models R_1 \equiv R_2 & \Leftrightarrow & R_1^\mathcal{I} = R_2^\mathcal{I} \\ \mathcal{I} \models R_1 \sqsubseteq R_2 & \Leftrightarrow & R_1^\mathcal{I} \subseteq R_2^\mathcal{I} \\ \mathcal{I} \models \mathrm{Trans}(R) & \Leftrightarrow & \\ \multicolumn{3}{r}{(\forall a,b,c \in \Delta^\mathcal{I})(\langle a,b\rangle \in R^\mathcal{I} \wedge \langle b,c\rangle \in R^\mathcal{I} \rightarrow \langle a,c\rangle \in R^\mathcal{I})} \end{array}$$

- $\mathcal{I} \models T$ kiolvasása: $\mathcal{I}$ kielégíti a T axiómát, ill. $\mathcal{I}$ modellje T-nek.
- Legyen $\mathcal{T}$ egy T-doboz (axiómák egy halmaza)
  - $\mathcal{I} \models \mathcal{T}$ ($\mathcal{I}$ modellje $\mathcal{T}$-nek) $\Leftrightarrow$ ha $\mathcal{T}$ minden axiómájának modellje, azaz minden $T \in \mathcal{T}$ esetén $\mathcal{I} \models T$
  - Egy $\mathcal{T}$ T-doboz szemantikai következménye a T állítás: $\mathcal{T} \models T \Leftrightarrow$ ha $\mathcal{T}$minden modellje kielégíti T-t, azaz minden olyan $\mathcal{I}$ esetén, melyre $\mathcal{I} \models \mathcal{T}$, fennáll, hogy $\mathcal{I} \models T$

Tartalom

Példa T-dobozra

- Családi kapcsolatok fogalomrendszerét leíró T-doboz:

$$
\begin{aligned}
\text{Nő} &\equiv \text{Ember} \sqcap \text{Nőnemű} \\
\text{Férfi} &\equiv \text{Ember} \sqcap \neg\text{Nő} \\
\text{Anya} &\equiv \text{Nő} \sqcap \exists\text{gyereke.Ember} \\
\text{Apa} &\equiv \text{Férfi} \sqcap \exists\text{gyereke.Ember} \\
\text{Szülő} &\equiv \text{Apa} \sqcup \text{Anya} \\
\text{Nagyanya} &\equiv \text{Anya} \sqcap \exists\text{gyereke.Szülő} \\
\text{SokgyerekesAnya} &\equiv \text{Anya} \sqcap (\geqslant 3\,\text{gyereke}) \\
\text{FiúsAnya} &\equiv \text{Anya} \sqcap \forall\text{gyereke.}\neg\text{Nő} \\
\text{Feleség} &\equiv \text{Nő} \sqcap \exists\text{férje.Férfi}
\end{aligned}
$$

Következtetési feladatok osztályozása

- A legegyszerűbb feladat:
  - **Konzisztencia**: egy $\mathcal{T}$ T-doboz konzisztens, ha van modellje, azaz létezik olyan $\mathcal{I}$, hogy $\mathcal{I} \models \mathcal{T}$.
- Alapvető következtetési feladatok T-dobozokon
  - **Kielégíthetőség**: egy C fogalom kielégíthető a $\mathcal{T}$ terminológia felett, ha létezik $\mathcal{T}$-nek olyan $\mathcal{I}$ modellje, hogy $C^{\mathcal{I}}$ nem üres.
  - **Tartalmazás (Alárendeltség)**: Egy C fogalmat tartalmaz egy D fogalom a $\mathcal{T}$ T-doboz felett, ha $\mathcal{T} \models C \sqsubseteq D$, azaz $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ teljesül $\mathcal{T}$ $\forall\, \mathcal{I}$ modelljére. Alternatíve: $C \sqsubseteq_{\mathcal{T}} D$
  - **Ekvivalencia**: A C és D fogalmak ekvivalensek a $\mathcal{T}$ terminológia felett, ha $\mathcal{T} \models C \equiv D$, azaz $C^{\mathcal{I}} = D^{\mathcal{I}}$ teljesül $\mathcal{T}$ minden $\mathcal{I}$ modelljében. Alternatív jelölés: $C \equiv_{\mathcal{T}} D$.
  - **Diszjunktság**: Két fogalom diszjunkt a $\mathcal{T}$ terminológia felett, ha $\mathcal{T} \models C \sqcap D \equiv \bot$, azaz $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ teljesül $\mathcal{T}$ minden $\mathcal{I}$ modelljére.
- Egy összetettebb feladat:
  - **Koherencia**: egy $\mathcal{T}$ T-doboz koherens, ha minden $\mathcal{T}$-ben előforduló A *atomi* fogalom kielégíthető $\mathcal{T}$ felett.

Következtetési feladatok – példák

- Példák: ha $\mathcal{T}$ a családi T-doboz, akkor az alábbi állítások igazak:
  - Nő $\sqcap$ Férfi nem kielégíthető a $\mathcal{T}$ T-doboz felett. (*)
  - $\mathcal{T} \models$ Nő $\sqsubseteq$ Ember (tartalmazás)
  - $\mathcal{T} \models$ Szülő $\equiv$ Ember $\sqcap$ $\exists$gyereke.Ember (ekvivalencia)
  - $\mathcal{T} \models$ Nő $\sqcap$ Férfi $\equiv \emptyset$ (diszjunktság)
- Bizonyos T-doboz feletti következtetési feladatok visszavezethetők üres T-doboz feletti következtetésre, pl. (*) helyett vizsgáljuk: Ember $\sqcap$ Nőnemű $\sqsubseteq$ Ember

Következtetések egymásra való visszavezetése

- Ha van egy módszerünk a tartalmazás eldöntésére ($\mathcal{AL}$ esetén is alkalmazható):
  - C kielégíthetetlen $\iff$ C része $\bot$ -nak
  - C és D ekvivalens $\iff$ C része D-nek és D része C-nek
  - C és D diszjunkt $\iff$ $C \sqcap D$ része $\bot$-nak
- Ha van egy módszerünk a kielégíthetőség eldöntésére (csak $\mathcal{ALC}$-től kezdve alkalmazható)
  - C része D-nek $\iff$ $C \sqcap \neg D$ kielégíthetetlen
  - C és D ekvivalens $\iff$ $(C \sqcap \neg D)$ és $(D \sqcap \neg C)$ is kielégíthetetlen
  - C és D diszjunkt $\iff$ $C \sqcap D$ kielégíthetetlen
- A kielégíthetőséget egyszerűbb vizsgálni (ebben csak egy fogalom van, míg a többi következtetési feladatban kettő)
- Ezért a különböző kifejezőerejű leíró logikai nyelvek esetén más-más következtetési feladatra érdemes algoritmust fejleszteni:
  - $\mathcal{AL}$ esetén: tartalmazás-vizsgálati algoritmus (ún. structural subsumption algorithm)
  - $\mathcal{ALC}$ és erősebb nyelvek esetén: kielégíthetőség-vizsgálati algoritmusok (tabló-algoritmus)

Következtetések egymásra való visszavezetése – példák

- Az „Apa része Szülő” feladat átfogalmazásai:
  - $\neg$Szülő és Apa diszjunktak.
  - $\neg$Szülő $\sqcap$ Apa ekvivalens a $\perp$ fogalommal.
  - $\neg$Szülő $\sqcap$ Apa kielégíthetetlen.
- A „Szülő ekvivalens Apa $\sqcup$ Anya” átalakításai:
  - A Szülő $\sqcap \neg$Apa $\sqcap \neg$Anya, Apa $\sqcap \neg$Szülő és Anya $\sqcap \neg$Szülő fogalmak mind kielégíthetetlenek.
  - Szülő része Apa $\sqcup$ Anya és Apa $\sqcup$ Anya része Szülő
  - $\neg$Szülő és Apa $\sqcup$ Anya diszjunktak, valamint Szülő és $\neg$Apa $\sqcap \neg$Anya is diszjunktak

Axiómák osztályozása

- **Definíciós axiómák.** Egy definíciós axióma segítségével egy új fogalmat vezetünk be
  - példa: Anya $\equiv$ Ember $\sqcap$ Nőnemű $\sqcap \exists$gyereke.Ember
  - a baloldalon egyetlen fogalomnév áll: ún. *elnevezett* fogalom
  - egy elnevezett fogalom pontosan egy axióma baloldalán fordulhat csak elő.
- **Háttértudást leíró axiómák.** A formalizálni kívánt terület ismereteit, háttértudását írják le.
  - példa: Doktorandusz $\sqsubseteq$ ($\geqslant 2$ nyelvvizsgája)
  - A háttértudásról szóló axiómákra nincs formai megkötés, tetszőleges C és D melett: $C \equiv D$ vagy $C \sqsubseteq D$ alakúak lehetnek
  - $C \equiv D$ átalakítható két tartalmazási axiómává: $C \sqsubseteq D$ és $D \sqsubseteq C$
- Háttértudást leíró axióma másik neve: **általános fogalomtartalmazási axióma** (**General Concept Inclusion axiom**, rövidítve GCI).

Ciklikus és ciklusmentes terminológiák

- Vizsgáljuk a T-doboz fogalomdefiníciós részét (csak egyenlőségek, baloldalon fogalomnevek, mindegyik egyszer)
  - Elnevezett fogalom: a baloldalon előforduló fogalomnév
  - Alapfogalom a többi, nem elnevezett fogalomnév
- Egyértelműen definiált terminológia (definitorial terminology)
  - Az alapfogalmak jelentése egyértelműen meghatározza az elnevezett fogalmak jelentését
  - A családi T-doboz ilyen, az alapfogalmak: Ember és Nőnemű az elnev. fogalmak: Nő , Férfi , Anya , Apa , Szülő , Nagyanya , ...
  - Alapinterpretáció: csak az alapfogalmakat definiálja
  - Egyértelműen definiált terminológia esetén elegendő az alapinterpretációt megadni
- Ciklikus terminológiák
  - Előfordulhat, hogy egy fogalom definíciója során felhasználjuk a definiálandó fogalmat:
    $$\text{Ember} \equiv \text{Állat} \sqcap \forall\text{szülője.Ember}$$
  - Ennél természetesen bonyolultabb esetek is lehetnek, amikor nem közvetlen a rekurzió!

Ciklikus terminológiák – fixpont szemantika

- Klasszikus példa: férfi csak férfi leszármazottal: Man with Only Male Offsprings (Momo)

$$\begin{aligned}
\text{Momo} &\equiv \text{Férfi} \sqcap \forall\text{gyereke.Momo}\\
\Delta &= \{Charles_1, Charles_2, \ldots\} \cup \{James_1, \ldots, James_{Last}\},\\
\text{Férfi}^{\mathcal{I}} &= \Delta,\\
\text{gyereke}^{\mathcal{I}} &= \{(Charles_i, Charles_{(i+1)}) | i \geq 1\} \cup\\
&\quad \cup\{(James_i, James_{(i+1)}) | 1 \leq i < Last\}.
\end{aligned}$$

- A ciklikus definíció egyenletként fogható fel: Momo $\equiv f$(Momo) ahol $f(X) =$ Férfi $\sqcap \forall$gyereke.X
- Az egyenletnek két fixpontja van: az egyik szerint az összes objektum Momo (legnagyobb fixpont – greatest fixpoint), míg a másik szerint csak a *James*-ek Momo-k (legkisebb fixpont – least fixpoint).
- A ciklikus terminológiák általában nem egyértelműen definiáltak, de pl. a legkisebb fixpont szemantika lerögzítésével azzá tehetők.

Ciklikus terminológiák – példa négy fixponttal

- Az az ember boldog ember, akinek minden barátja boldog ember:

$$\text{BoldogEmber} \equiv \text{Ember} \sqcap \forall \text{barátja}.\text{BoldogEmber}$$

- Alapfogalom: Ember, elnevezett fogalom: BoldogEmber
- Példa-interpretáció:
  $\mathcal{I}$ alapinterp.:
  $\Delta^{\mathcal{I}} = \{a, b, c, d\}$, $\text{Ember} = \Delta^{\mathcal{I}}$, $\text{barátja}^{\mathcal{I}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle c, d\rangle, \langle d, c\rangle\}$
- A BoldogEmber elnevezett fogalom lehetséges értelmezései (fixpontok):
  - $\text{BoldogEmber}^{\mathcal{I}} = \{a, b\}$ (folytonos vonallal határolt rész)
  - $\text{BoldogEmber}^{\mathcal{I}} = \{c, d\}$ (szaggatott vonallal határolt rész)
  - $\text{BoldogEmber}^{\mathcal{I}} = \{a, b, c, d\}$ (teljes alaphalmaz)
  - $\text{BoldogEmber}^{\mathcal{I}} = \{\}$ (üres halmaz)

Ciklusmentes terminológiák

- Egy terminológia ciklusmentes, ha nem ciklikus.
- Egy T-doboz ciklusmentes fogalomdefiníciós része kiküszöbölhető:
  - Egy ciklusmentes T-doboz kiterjesztése:
    a jobboldalon levő elnevezett fogalmakat helyettesítjük a def.-jukkal
  - Így a T-dobozban elnevezett fogalmak csak a baloldalon lehetnek
- Példa: A családi kapcsolatokat leíró T-doboz kiterjesztése

$$\begin{aligned}
\text{Nő} &\equiv \text{Ember} \sqcap \text{Nőnemű} \\
\text{Férfi} &\equiv \text{Ember} \sqcap \neg(\text{Ember} \sqcap \text{Nőnemű}) \\
\text{Anya} &\equiv \text{Ember} \sqcap \text{Nőnemű} \sqcap \exists \text{gyereke}.\text{Ember} \\
\text{Apa} &\equiv (\text{Ember} \sqcap \neg(\text{Ember} \sqcap \text{Nőnemű})) \sqcap \exists \text{gyereke}.\text{Ember} \\
\text{Szülő} &\equiv ((\text{Ember} \sqcap \neg(\text{Ember} \sqcap \text{Nőnemű})) \sqcap \exists \text{gyereke}.\text{Ember}) \sqcup \\
&\ \sqcup(\text{Ember} \sqcap \text{Nőnemű} \sqcap \exists \text{gyereke}.\text{Ember}) \\
\text{Nagyanya} &\equiv (\text{Ember} \sqcap \text{Nőnemű} \sqcap \exists \text{gyereke}.\text{Ember}) \sqcap \\
&\ \sqcap\exists \text{gyereke}.(((\text{Ember} \sqcap \neg(\text{Ember} \sqcap \text{Nőnemű})) \sqcap \\
&\qquad \exists \text{gyereke}.\text{Ember}) \sqcup \\
&\qquad \sqcup(\text{Ember} \sqcap \text{Nőnemű} \sqcap \exists \text{gyereke}.\text{Ember})) \\
&\ldots
\end{aligned}$$

Ciklusmentes T-doboz kiküszöbölése

- Ciklusmentes T-doboznál a következtetési problémák visszavezethetők az üres T-doboz esetére.
  - Példa: Férfi és Nő diszjunktságának eldöntése: $\text{Nő} \sqcap \text{Férfi}$
    - Az elnevezett fogalmakat a kiterjesztett T-doboz szerinti definícióval helyettesítjük, pl. $\text{Nő} \sqcap \text{Férfi} \rightsquigarrow$ $\text{Ember} \sqcap \text{Nőnemű} \sqcap \text{Ember} \sqcap \neg(\text{Ember} \sqcap \text{Nőnemű})$
    - Így a T-dobozt már el is hagyhatjuk, hiszen egy $\mathcal{T}$-ben definiált elnevezett fogalom sem szerepel vizsgált fogalomban
- Egy C fogalom *kiterjesztése* egy $\mathcal{T}$ ciklusmentes T-doboz szerint:
  - Előállítjuk $\mathcal{T}$ kiterjesztését, nevezzük ezt $\mathcal{T}$'-nek
  - A C-ben előforduló elnevezett fogalmakat helyettesítjük $\mathcal{T}$'-beli definíciójuk jobboldalával
- ha egy C fogalom kielégíthetőségét vizsgáljuk egy $\mathcal{T}$ ciklusmentes T-doboz felett
  elegendő (és szükséges) C kiterjesztésének kielégíthetőségét vizsgálni *üres T-doboz* felett

Általános T-dobozok kezelése a következtetési feladatokban

- (Ismétlés): A ciklusmentes fogalomdefiníciós axiómák a kiterjesztés módszerével kiküszöbölhetőek
- Általános fogalomtartalmazási axiómák ($C \sqsubseteq D$) kezelése
  - (Ismétlés: $C \equiv D$ helyettesíthető: $\{C \sqsubseteq D, D \sqsubseteq C\}$-vel).
  - $C \sqsubseteq D$ helyettesíthető a $\top \sqsubseteq \neg C \sqcup D$ állítással
  - egy tetszőleges $\{C_1 \sqsubseteq D_1, C_2 \sqsubseteq D_2, \ldots, C_n \sqsubseteq D_n\}$ T-doboz átalakítható egyetlen, vele ekvivalens $\top \sqsubseteq C_{\mathcal{T}}$ alakú axiómává, ahol:

$$C_{\mathcal{T}} = (\neg C_1 \sqcup D_1) \sqcap (\neg C_2 \sqcup D_2) \sqcap \cdots \sqcap (\neg C_n \sqcup D_n).$$

  - A $C_{\mathcal{T}}$ fogalom a $\mathcal{T}$ T-doboz *belsősítése* (internalisation)
    - Egy $\mathcal{I}$ interpretáció pontosan akkor modellje egy $\mathcal{T}$ T-doboznak ($\mathcal{I} \models \mathcal{T}$), ha az interpretáció alaphalmazának minden eleme a $C_{\mathcal{T}}$ belsősített fogalom példánya
    - Explicit megvalósítás: a tabló-algoritmusban minden csúcshoz hozzávesszük a $C_{\mathcal{T}}$ címkét.
    - Implicit megvalósítás ($\mathcal{SH}$ vagy ennél bővebb nyelv esetén): a fogalmi axiómákat „beleolvasztjuk” a kielégítendő fogalomba és új szerepaxiómákba

Terminológiák belsősítése (internalization) $\mathcal{SH}$-ban

- Alapgondolat: egy új, $C_\mathcal{T}$-t is tartalmazó fogalom kielégíthetőségét vizsgáljuk egy fogalmi axiómákat nem, de további szerepaxiómákat tartalmazó új T-doboz felett
- $\mathcal{SH}$ nyelven egy C fogalom egy $\mathcal{T}$ T-doboz feletti kielégíthetőség-vizsgálata helyettesíthető egy

$$C_{C,\mathcal{T}} = C \sqcap C_\mathcal{T} \sqcap \forall U.C_\mathcal{T}$$

  fogalom kielégíthetőség-vizsgálatával egy $\mathcal{T}_{C,\mathcal{T}}$ T-doboz felett
- Ehhez a szerepnevek halmazát egy új U (ún. szimulált univerzális) szereppel bővítjük ki, ez minden szerepnél $\sqsupseteq$ tranzitív reláció
- A $\mathcal{T}_{C,\mathcal{T}}$ T-doboz fogalmi axiómákat nem tartalmaz, szerepaxiómái:
  - **Trans**(U)
  - Új szereptartalmazási axiómák: $R \sqsubseteq U$, $\mathrm{Inv}(R) \sqsubseteq U$, minden olyan R szerepre, amely C-ben vagy $\mathcal{T}$-ben előfordul
- Állítás: C kielégíthető $\mathcal{T}$ felett, csakkor ha $C_{C,\mathcal{T}}$ kielégíthető $\mathcal{T}_{C,\mathcal{T}}$ felett
- Ezt a módszert nevezik belsősítésnek (internalization).

Tartalom

Az A-doboz

- A világban jelenlevő objektumok reprezentálására egy új névfajtát vezetünk be, az *egyedneveket*. jelölésük, a, b, c stb.
- Az adatdoboz (A-doboz) adatállításokat tartalmaz, ezek lehetnek:
  - fogalmi állítások: $C(a)$, pl. Apa(PÉTER), ($\exists$állása.$\top$)(PÉTER)
  - szerepállítások: $R(a,b)$, pl. (szülője$^-$)(PÉTER, PÁL).
- $\mathcal{I}$ interpretációs függvényt ki kell bővíteni: minden a egyednévhez $\mathcal{I}$ hozzárendel egy neki megfelelő $a^\mathcal{I} \in \Delta^\mathcal{I}$ elemet
- $\mathcal{I}$ kielégíti a $C(a)$ fogalmi állítást ($\mathcal{I} \models C(a)$) csakkor, ha $a^\mathcal{I} \in C^\mathcal{I}$,
- $\mathcal{I}$ kielégíti a $R(a,b)$ szerepállítást ($\mathcal{I} \models R(a,b)$) csakkor, ha $\langle a^\mathcal{I}, b^\mathcal{I} \rangle \in R^\mathcal{I}$.
- Egyedi nevek feltételezése (UNA – Unique Name Assumption) – nem mindig szükséges
  - UNA = elvárjuk azt, hogy az egyednevek értelmezése páronként különböző legyen.

Következtetések A-dobozon

- Az A-doboz hasonlít egy relációs adatbázisra, amelyben csak egy- és kétoszlopú táblák vannak. De az adatbázisoknál megszokott „zárt világ szemantika” helyett az A-dobozra a „nyílt világ szemantika” jellemző: a tudásbázis nem teljes, amit nem tudunk (nincs benne explicit módon az A-dobozban) az nem feltétlenül hamis!
- A-doboz konzisztencia
  - Egy $\mathcal{A}$ A-doboz akkor konzisztens egy $\mathcal{T}$ T-doboz felett, ha létezik egy olyan $\mathcal{I}$ interpretáció, amely modellje $\mathcal{A}$-nak és $\mathcal{T}$-nek egyszerre. Például, az {Anya(MARI),Apa(MARI)} A-doboz konzisztens az üres T-doboz felett, viszont inkonzisztens a családi kapcsolatokat leíró T-doboz felett.
  - Egy ciklusmentes T-doboz feletti A-doboz-következtetések visszavezethetők egy kiterjesztett A-dobozon való következtetésre. (A ciklusmentes T-doboz itt is kiküszöbölhető).
- Definíció: $\mathcal{A} \models_\mathcal{T} \alpha$: Az $\mathcal{A}$ A-dobozból a $\mathcal{T}$ T-doboz felett következik az α állítás: ha minden $\mathcal{A}$-t és $\mathcal{T}$-t kielégítő interpretáció ($\mathcal{A}$ és $\mathcal{T}$ minden közös modellje), biztosan kielégíti α-t.

Következtetések A-dobozon

- További következtetési feladatok A-dobozokra
  - *Példányvizsgálat* (*instance check*): egy α adatállítás következménye-e egy $\mathcal{A}$ adatdoboznak egy $\mathcal{T}$ felett ($\mathcal{A} \models_{\mathcal{T}} \alpha$)? Példa: igaz-e, hogy Apa(MIKLÓS), a családi T-doboz felett? Visszavezetés konzisztencia-vizsgálatra:

$$\mathcal{A} \models_{\mathcal{T}} C(a) \iff \mathcal{A} \cup \{\neg C(a)\} \text{ nem konzisztens } \mathcal{T} \text{ felett}$$

  - *Példánykikeresés* (*instance retrieval*): adott C fogalomkifejezés (vagy R szerepkifejezés) esetén meg kell állapítani, hogy mely egyednevek (névpárok) tartoznak *biztosan* bele. Példa: mik a példányai az Ember $\sqcap \neg$Nőnemű fogalomnak?
  - *Egyed-realizáció* (*realisation*): egy adott egyedhez meg kell keresni a legszűkebb olyan fogalma(ka)t, amelynek biztosan példánya (több ilyen minimális fogalom is lehetséges).
- Tisztán terminológiai következtetés A-doboz következtetővel
  - *Fogalom-kielégíthetőség*:

$$C \text{ kielégíthető } (\mathcal{T} \text{ felett}) \iff \{C(a)\} \text{ adatdoboz konzisztens } (\mathcal{T} \text{ felett})$$

Nyílt és zárt világ szemantikák

- Tekintsünk egyetlen állítást: gyereke(PÉTER,PÁL)
- Hogyan változik az állítás jelentése, aszerint hogy
  - adatbázisban (RDBMS) vagy
  - A-dobozban szerepel?
- Ha ez egy adatbázis-tábla egyetlen sora
  - zárt világ szemantika szerinti jelentés:
  - Péternek egyetlen gyermeke van, Pál
- Ha ez egy A-doboz egyetlen állítása
  - nyílt világ szemantika szerinti jelentés:
  - Péternek van egy Pál nevű gyermeke (de lehet más gyereke is)
  - Ha azt is közölni szeretnénk, hogy Pál Péter egyetlen gyermeke, akkor pl. hozzátehetjük: ($\leqslant$ 1gyereke)(PÉTER).

Eset-szétválasztáson alapuló következtetés nyílt világban

- Egy klasszikus példa (nevezzük az alábbi adatdobozt $\mathcal{A}_{OI}$-nak):

  gyereke(IOKASZTÉ,OIDIPUSZ)
  gyereke(IOKASZTÉ,POLÜNEIKÉSZ)
  gyereke(OIDIPUSZ,POLÜNEIKÉSZ)
  gyereke(POLÜNEIKÉSZ,THERSZANDROSZ)

  Apagyilkos(OIDIPUSZ)

  $\neg$ Apagyilkos(THERSZANDROSZ)

- Az $\mathcal{A}_{OI}$ A-dobozra vonatkozóan az alábbi kérdést tesszük fel:

  Van-e Iokaszténak olyan gyermeke, aki apagyilkos, és akinek van egy gyermeke, aki nem apagyilkos?

  azaz:

  $\mathcal{A}_{OI} \models (\exists$gyereke.(Apagyilkos $\sqcap \exists$gyereke.$\neg$Apagyilkos))(IOKASZTÉ)?

- A válasz: igen, de a bizonyításhoz eset-szétválasztás szükséges!

$\mathcal{SHIQ}$ T- és A-dobozok elsőrendű logikában

- Minden C fogalomkifejezésnek megfeleltetünk egy $\Phi_C(x)$ elsőrendű állítást (egyetlen szabad változója x)
  - $\Phi_C(x)$ jelentése: x a C fogalom példánya
  - Pl. ha $C =$ Ember $\sqcap \neg$Nő $\sqcap \exists$gyereke.$\top$, akkor $\Phi_C(x) =$ Ember$(x) \wedge \neg$Nő$(x) \wedge \exists y.$gyereke(x, y)
- Minden R szerepkifejezésnek egy $\Phi_R(x, y)$ (két szabad változójú) állítást feleltetünk meg
  - $\Phi_R(x, y)$ jelentése: x és y R-kapcsolatban vannak
  - Pl. ha $R =$ gyereke$^-$ akkor $\Phi_R(x, y) =$ gyereke(y, x)
- Fogalomkifejezések átírása:

$$\begin{aligned}
\Phi_A(x) &= A(x)\\
\Phi_\top(x) &= \mathrm{TRUE}\\
\Phi_\bot(x) &= \mathrm{FALSE}\\
\Phi_{\neg C}(x) &= \neg\Phi_C(x)\\
\Phi_{C_1 \sqcap C_2}(x) &= \Phi_{C_1}(x) \wedge \Phi_{C_2}(x)\\
\Phi_{C_1 \sqcup C_2}(x) &= \Phi_{C_1}(x) \vee \Phi_{C_2}(x)
\end{aligned}$$

$\mathcal{SHIQ}$ elsőrendű logikában (2)

- Fogalomkifejezések átírása (folyt.)

$$
\begin{aligned}
\Phi_{\forall R.C}(x) &= \forall y.\,(\Phi_R(x,y) \rightarrow \Phi_C(y)) \\
\Phi_{\exists R.C}(x) &= \exists y.\,(\Phi_R(x,y) \wedge \Phi_C(y)) \\
\Phi_{(\geqslant n\,R.C)}(x) &= \exists y_1,\ldots,y_n.\,(\;\Phi_R(x,y_1) \wedge \cdots \wedge \Phi_R(x,y_n) \wedge \\
&\qquad \Phi_C(y_1) \wedge \cdots \wedge \Phi_C(y_n) \wedge \bigwedge_{i<j} y_i \neq y_j\;) \\
\Phi_{(\leqslant n\,R.C)}(x) &= \forall y_1,\ldots,y_{n+1}.\,(\;\Phi_R(x,y_1) \wedge \cdots \wedge \Phi_R(x,y_{n+1}) \wedge \\
&\qquad \Phi_C(y_1) \wedge \cdots \wedge \Phi_C(y_{n+1}) \rightarrow \bigvee_{i<j} y_i = y_j\;)
\end{aligned}
$$

- Szerepkifejezések átírása:

$$
\begin{aligned}
\Phi_{R_A}(x,y) &= R_A(x,y) \\
\Phi_{R_A^-}(x,y) &= R_A(y,x)
\end{aligned}
$$

$\mathcal{SHIQ}$ elsőrendű logikában (3)

- Terminológiai axiómák átírása:

$$
\begin{aligned}
\Phi_{C_1 \equiv C_2} &= \forall x.\,(\Phi_{C_1}(x) \leftrightarrow \Phi_{C_2}(x)) \\
\Phi_{C_1 \sqsubseteq C_2} &= \forall x.\,(\Phi_{C_1}(x) \rightarrow \Phi_{C_2}(x)) \\
\Phi_{R_1 \equiv R_2} &= \forall x,y.\,(\Phi_{R_1}(x,y) \leftrightarrow \Phi_{R_2}(x,y)) \\
\Phi_{R_1 \sqsubseteq R_2} &= \forall x,y.\,(\Phi_{R_1}(x,y) \rightarrow \Phi_{R_2}(x,y)) \\
\Phi_{\mathrm{Trans}(R)} &= \forall x,y,z.\,(\Phi_R(x,y) \wedge \Phi_R(y,z) \rightarrow \Phi_R(x,z))
\end{aligned}
$$

- Adataxiómák átírása:

$$
\begin{aligned}
\Phi_{C(a)} &= \Phi_C(a) \\
\Phi_{R(a_1,a_2)} &= \Phi_R(a_1,a_2)
\end{aligned}
$$

$\mathcal{SHIQ}$ elsőrendű logikában – példa

- Példa T-doboz

$$
\begin{aligned}
\mathcal{T} = \{&\text{LányosApa} \equiv \text{Ember} \sqcap \neg\text{Nő} \sqcap \forall\text{gyereke.Nő} \sqcap \exists\text{gyereke.}\top, \\
&\text{LányosApa} \sqsubseteq \text{Boldog}\}
\end{aligned}
$$

- A példa elsőrendű megfelelője

$$
\begin{aligned}
&\forall x.\,(\text{LányosApa}(x) \leftrightarrow \\
&\qquad \text{Ember}(x) \wedge \neg\text{Nő}(x) \\
&\qquad \wedge \forall y.\,(\text{gyereke}(x,y) \rightarrow \text{Nő}(y)) \\
&\qquad \wedge \exists y.\text{gyereke}(x,y)) \\
&\wedge \forall x.\,(\text{LányosApa}(x) \rightarrow \text{Boldog}(x))
\end{aligned}
$$

Tartalom

Konkrét tartományok: a (D) nyelvkiterjesztés

- Nagykorú (ember) fogalma: 18 évnél idősebb ember.
- Kisérlet $\mathcal{SHIQ}$-beli megfogalmazásra:

$$\begin{array}{rcl} \text{Nagykorú} & \equiv & \text{Ember} \sqcap \exists\text{életkora.FelnőttKor} \\ \text{FelnőttKor} & \sqsubseteq & \text{Életkor} \\ \text{Életkor} & \sqsubseteq & \text{TermészetesSzám} \end{array}$$

- Ez kényelmetlen, pontatlan, jobb lenne ha az életkora szerep értékkészlete a természetes számok egy részhalmaza lehetne.
- Megoldás: a nyelv bővítése konkrét tartományokkal (adattípusokkal), pl. $\mathcal{SHIQ}(\mathbf{D})$
  - Például egy konkrét tartomány lehet a természetes számok
  - Új szimbolumok (szintaktikus elemek): adattípus-jel, pl.

$$\mathbf{D} = \{\text{intv}_{i,j} \mid i \leq j \text{ természetes szám}\}$$

  - A jelek szemantikája:

$$\text{intv}_{i,j}^{\mathbf{D}} = \{k \mid i \leq k \leq j\}$$

Konkrét tartományok: a (D) nyelvkiterjesztés (2)

- Konkrét alaphalmaz: $\Delta_{\mathbf{D}} = \{d^{\mathbf{D}} \mid d \in \mathbf{D}\}$ – a példában = természetes számok
- Az absztrakt szerepek mellett lesznek konkrét szerepeink, $\Delta^{\mathcal{I}}$ és $\Delta_{\mathbf{D}}$ között, pl. életkora
- Egy R_D konkrét szerep csak $\exists R_D.dk$ vagy $\forall R_D.dk$ alakban fordulhat elő,
- dk egy olyan konkrétfogalom-kifejezés, amely $d \in \mathbf{D}$ adattípusokból az unió, metszet és negáció segítségével épül fel
- Példák
  - $\text{Nagykorú} \equiv \exists\text{életkora.intv}_{18,120}$
  - $\text{TanuóVagyNyugdíjasKorú} \equiv \exists\text{életkora.}(\text{intv}_{0,18} \sqcup \text{intv}_{62,120})$

Egyedfogalmak

- Egyedfogalom (nominal): olyan fogalom, amelynek egyetlen példánya lehet. Jelölése: $\mathcal{O}$, pl. $\mathcal{SHIQ}$ + egyedfogalom = $\mathcal{SHOIQ}$
- Példa: földrajzi ontológia: Kontinens, Ország stb. fogalmak, helye szerep, $\text{EurópaiOrszág} \equiv \exists\text{helye.Európa}$. – itt Európa egy egyedfogalom.
- Miért nem lehet Európa egy általános fogalom?
  - Próbáljuk szimulálni $\mathcal{SHIQ}$-ban:

$$\begin{array}{rcl} \text{Kontinens} & \equiv & \text{Európa} \sqcup \text{Ázsia} \sqcup \text{Amerika} \sqcup \cdots \\ \text{Európa} \sqcap \text{Ázsia} & \sqsubseteq & \bot \\ \text{Európa} \sqcap \text{Amerika} & \sqsubseteq & \bot \\ \ldots & & \text{(a kontinensek páronként diszjunktak)} \end{array}$$

  - Definiáljuk a következő – diszjunktnak gondolt – fogalmakat:

$$\begin{array}{rcl} \text{ÓriásOrszág} & \equiv & (\geqslant 2\,\text{helye.Kontinens}) \\ \text{EUOrszág} & \sqsubseteq & \forall\text{helye.Európa} \end{array}$$

  - ÓriásOrszág és EUOrszág diszjunktságának bizonyításához tudnunk kell, hogy Európa egyedfogalom (egyetlen példánya van).

Egyedfogalmak (2)

- Egyedfogalmak jelölése deklarációval: Indiv(Európa) (nem szokásos)
- Egyedfogalmak jelölése használatkor: kapcsos zárójelbe tett egyednév (vö. adatdobozok)
- Példa: $\text{EurópaiVállalat} \equiv \forall\text{telephelye.}\forall\text{helye.}\{\text{EURÓPA}\}$
- Az egyedfogalom általánosítása (szintaktikus édesítőszer) enumerációs fogalom: $\{a, b, \ldots u\} \Longrightarrow \{a\} \sqcup \{b\} \sqcup \ldots \sqcup \{u\}$
- Példa: $\text{Eurázsia} \equiv \{\text{EURÓPA,ÁZSIA}\} \Longrightarrow \text{Eurázsia} \equiv \{\text{EURÓPA}\} \sqcup \{\text{ÁZSIA}\}$

További nyelvkiterjesztések

- Szerepkonstruktorok

Elnevezés	Szintaxis	Szemantika
Univerzális szerep	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Metszet	$R_1 \sqcap R_2$	$R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}}$
Unió	$R_1 \sqcup R_2$	$R_1^{\mathcal{I}} \cup R_2^{\mathcal{I}}$
Komplemens	$\neg R$	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}$
Kompozíció	$R_1 \circ R_2$	$R_1^{\mathcal{I}} \circ R_2^{\mathcal{I}}$
Tranzitív lezárás	R^+	$\bigcup_{n \geq 1} (R^{\mathcal{I}})^n$
Reflexív-tranzitív lezárás	R^*	$\bigcup_{n \geq 0} (R^{\mathcal{I}})^n$
Szerepszűkítés	$R\|_C$	$R^{\mathcal{I}} \sqcap (\Delta^{\mathcal{I}} \times C^{\mathcal{I}})$
Azonosság	$id(C)$	$\{\langle d, d \rangle \mid d \in C^{\mathcal{I}}\}$

- Példa:

$$\text{nagyanyja} \equiv \text{szülője} \circ \text{anyja}$$

Szerepkonstruktorok – példák

$$
\begin{aligned}
\text{anyja} &\equiv \text{szülője}|_{\text{Nőnemű}} \\
\text{testvére} &\equiv (\text{szülője} \circ \text{gyereke}) \sqcap \neg id(\top) \\
\text{fia} &\equiv \text{gyereke}|_{\neg\text{Nőnemű}} \\
\text{őse} &\equiv \text{szülője}^+ \\
\text{őseVagyMaga} &\equiv \text{szülője}^* \\
\text{vérrokona} &\equiv (\text{őseVagyMaga} \circ \text{őseVagyMaga}^-) \sqcap \neg id(\top)
\end{aligned}
$$

További nyelvkiterjesztések (2)

- Szerepérték-leképezések (role value maps):

$$R \subseteq S \qquad \text{és} \qquad R = S$$

- Jelentésük:

$$
\begin{aligned}
(R \subseteq S)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.\langle a, b \rangle \in R^{\mathcal{I}} \rightarrow \langle a, b \rangle \in S^{\mathcal{I}}\} \\
(R = S)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.\langle a, b \rangle \in R^{\mathcal{I}} \leftrightarrow \langle a, b \rangle \in S^{\mathcal{I}}\}
\end{aligned}
$$

- Példák:
  - (szülője $\subseteq$ barátja) azon gyerekek, akiknek minden szülője egyben barátja is
  - (szülője $=$ barátja) azok a gyerekek, akik a szüleikkel és csak a szüleikkel barátkoznak
  - (barátja $\subseteq$ szülője $\circ$ ismerőse) azok a gyerekek, akiknek minden barátját valamelyik szülőjük ismeri
- Sajnos a fejlettebb szerepkonstruktorok és a szerepérték-leképezések eldönthetetlenné teszik a logikát

A szerepkompozíció és az eldönthetőség

- A szerepkompozíció korlátozott, hogy a logika eldönthető maradjon
- Meg kell tiltani a ciklikus axiómákat, pl. $S_1 \circ \ldots \circ R \circ \ldots \circ S_n \sqsubseteq R$
- A ciklusok tiltására megkövetelhetjük, hogy
  - legyen egy $\prec$ erős (nem reflexív) részbenrendezés a szerepneveken
  - $S_1 \circ \ldots \circ S_n \sqsubseteq R$ axióma esetén $S_i \prec R$ fennálljon, minden i-re
- De a ciklikus szerepkompozíció speciális esetei fontosak lehetnek, pl.

$$\text{tulajdona} \circ \text{része} \sqsubseteq \text{tulajdona}$$

a tulajdon része is a tulajdonos birtokában van

$$\text{helye} \circ \text{tartalmazója} \sqsubseteq \text{helye}$$

a betegség helyének tartalmazója is a betegség helyének tekintendő

- Ha megengedjük az $S \circ R \sqsubseteq S$ alakú axiómákat, akkor az $R' \circ S' \sqsubseteq S'$ formájúakat is meg kell engedni (az első mindkét oldalát invertálva a másodikat kapjuk, mert $\text{Inv}(S \circ R) = \text{Inv}(R) \circ \text{Inv}(S)$), pl.

$$\text{része}^- \circ \text{tulajdona}^- \sqsubseteq \text{tulajdona}^- \rightsquigarrow$$

$$\text{tartalmazója} \circ \text{tulajdonosa} \sqsubseteq \text{tulajdonosa}$$

A $\mathcal{RIQ}$ nyelv

- A $\mathcal{RIQ}$ nyelv: a $\mathcal{SHIQ}$ nyelvet kiterjeszti $S \circ R \sqsubseteq S$ és $R \circ S \sqsubseteq S$ alakú axiómákkal, de a ciklusmentesség biztosítására kiköti, hogy ezekben – és $R \sqsubseteq S$ esetén is – $R \prec S$ legyen, ahol $\prec$ egy erős részbenrendezés
- A $\mathcal{RIQ}$ nyelv eldönthető
- Egyes szerzők szerint a (szereptartalmazási szempontból) ciklikus T-doboz nem is lehet értelmes (modellezési hiba)
- Vegyük észre, hogy a $\mathcal{RIQ}$ nyelven nem lehet szerepegyenlőségi axiómát megfogalmazni
- Ez nem gond, mert ekvivalens következtetési feladatot kapunk, ha az egyenlő szerepnevek halmazából kiválasztunk egy reprezentánst, és ezzel helyettesítjük az összes vele egyenlő szerepnevet a T- és A-dobozban

A $\mathcal{SROIQ}$ nyelv

- Egy $w \sqsubseteq R$ szereptartalmazási axióma (role inclusion axiom – RIA) $\prec$-reguláris – ahol R atomi szerep és '$\prec$' erős részbenrendezés – ha:
  - $w = R \circ R$ (R tranzitív); vagy
  - $w = R^-$ (R szimmetrikus); vagy
  - $w = S_1 \circ \ldots \circ S_n$, és $S_i \prec R$, minden $1 \leq i \leq n$-re; vagy
  - $w = R \circ S_1 \circ \ldots \circ S_n$, és $S_i \prec R$, minden $1 \leq i \leq n$-re; vagy
  - $w = S_1 \circ \ldots \circ S_n \circ R$, és $S_i \prec R$, minden $1 \leq i \leq n$-re.
- A $\mathcal{SROIQ}$ nyelv: a $\mathcal{ALCOIQ}$ nyelv kiterjesztése a következőkkel:
  - Fogalmak: a $\exists R$.Self fogalomkifejezés, jelentése:$\{x \in \Delta | \langle x, x \rangle \in R^I\}$; például Nárcisztikus $\equiv$ $\exists$szereti.Self
  - Szerepek: az U univerzális szerep
  - T-doboz: egy valamilyen $\prec$-re nézve reguláris RIA-k halmaza, plusz:
    - Ref(R): Az R szerep reflexív
    - Irr(R): Az R szerep irreflexív
    - Dis(R, S): Az R és S szerepek diszjunktak
  - A-doboz: ($\mathcal{ALCOIQ}$ A-doboz plusz) negált szerepállítás, pl. $\neg$barátja(A, B)

A $\mathcal{SROIQ}$ nyelv – egyszerű szerepek

- Informális definíció: $\mathcal{SROIQ}$-ban egy szerep egyszerű, ha nem tartalmaz szerepkompozícióval előállítható szerepet
- Definíció: Legyen $\mathcal{R}$ RIA-k egy halmaza. Az $\mathcal{R}$-re nézve egyszerű szerepek halmazát induktív módon definiáljuk:
  - egy R atomi szerep egyszerű, ha nem fordul elő egyetlen $\mathcal{R}$-beli RIA jobb oldalán sem;
  - egy R^- szerep egyszerű. ha R az;
  - egy $\mathcal{R}$-beli RIA jobb oldalán előforduló R egyszerű, ha minden $w \sqsubseteq R \in \mathcal{R}$ esetén $w = S$ alakú, ahol S egy egyszerű szerep.
- Az alábbi helyeken csak egyszerű szerep használható
  - R-doboz: Irr(S) és Dis(R, S) axiómákban
  - A-doboz: a negált szerepállításokban
  - T-doboz: a számosság-korlátozásokban (mint már $\mathcal{SHIQ}$-ban is), valamint a $\exists R$.Self fogalomkifejezésben
- Az így definiált $\mathcal{SROIQ}$ nyelv eldönthető, és ez szolgál az OWL2 matematikai alapjául.

Tartalom

A tárgyalt következtetési algoritmusok

- Ism.: minden következtetési feladat visszavezethető T-doboz/A-doboz kielégíthetőség-vizsgálatra ($\mathcal{ALC}$-től kezdve).
  (közös szóhasználat: A-doboz kielégíthető $\equiv$ A-doboz konzisztens)
- Legismertebb következtetési módszerek
  - *Strukturális alárendeltségi algoritmus:* két fogalomkifejezés szintaktikai struktúráját hasonlítja össze. Nagyon hatékony, de max. $\mathcal{ALN}$-ig jó. (Teljes negálást és uniót nem enged meg!)
  - *Tabló-algoritmus:* egy fogalomkifejezés vagy adatdoboz kielégíthetőségét vizsgálja, ezt igazoló modell építésével.
- Ebben a kurzusban:
  - struktúrális alárendeltségi alg. $\mathcal{AL}$-re
  - tabló-alg. $\mathcal{ALCN}$, $\mathcal{SHIQ}$ fogalomkifejezések kielégíthetőségére
  - tabló-alg. $\mathcal{SHIQ}$ A-dobozra
- Kitekintés: általános elsőrendű tételbizonyítók specializálhatók DL-re, pl.
  - A Vampire (Voronkov) elsőrendű tételbizonyító felhasználható ontológia-következtetésre (Horrocks, Voronkov)
  - DLog (Lukácsy, Szeredi, Zombori): PTTP (Prolog Technology Theorem Proving) alapú A-doboz következtető

Tartalom

Az algoritmushoz szükséges normálalak

- Az alap-algoritmus $\mathcal{FL}_0$ nyelvre (ebben csak $\sqcap$ és $\forall R.C$ megengedett)
  - Bevezetjük a vizsgálandó fogalomkifejezések alábbi normálalakját:
    $$A_1 \sqcap \cdots \sqcap A_m \sqcap \forall R_1.C_1 \sqcap \cdots \sqcap \forall R_n.C_n$$
    ahol $A_1, \ldots, A_m$ különböző fogalomnevek, $R_1, \ldots, R_n$ különböző szerepnevek, és $C_1, \ldots, C_n$ $\mathcal{FL}_0$ fogalomkifejezések normálalakban.
- Vizsgáljuk $C \sqsubseteq D$ fenállását, ahol C $\mathcal{FL}_O$ fogalomkifejezés normálalakja a fenti és D $\mathcal{FL}_O$ fogalomkifejezés normálalakja
  $$B_1 \sqcap \cdots \sqcap B_k \sqcap \forall S_1.D_1 \sqcap \cdots \sqcap \forall S_l.D_l$$
- $C \sqsubseteq D$ fennáll, ha a D metszet „szűkebb-egyenlő” C-nél, azaz D minden tagjához van C-ben egy annál $\sqsubseteq$ tag, pontosítva:
  (i) $\forall i, 1 \leq i \leq k, \exists j, 1 \leq j \leq m$ *amelyre* $B_i = A_j$
  (ii) $\forall i, 1 \leq i \leq l, \exists j, 1 \leq j \leq n$ *amelyre* $S_i = R_j$ *és* $C_j \sqsubseteq D_i$
- Az $\mathcal{FL}_O$-algoritmus kiterjeszthető $\mathcal{ALN}$-ig: az atomi negálás ($\neg A$), az egyszerűsített létezési korlátozás ($\exists R.\top$) és a számosság-korlátozások ($(\leqslant n\,R)$, $(\geqslant\ R)$) az atomi fogalmakhoz hasonlóan kezelhetők.
- De a strukturális alárendeltségi algoritmus nem képes kezelni az $\sqcup$, a teljes $\neg$ és a teljes $\exists R.C$ konstruktorokat.

A normálalak kiterjesztése $\mathcal{AL}$ nyelvre

- A normálalakú kifejezés lehet a $\bot$ jel, vagy
  $$A_1 \sqcap \cdots \sqcap A_k$$
  $$\sqcap \neg B_1 \sqcap \cdots \sqcap \neg B_l$$
  $$\sqcap \exists R_1.\top \sqcap \cdots \sqcap \exists R_m.\top$$
  $$\sqcap \forall S_1.C_1 \sqcap \cdots \sqcap \forall S_n.C_n,$$
- ahol
  - $A_1, \ldots, A_k, B_1, \ldots, B_l$ különböző fogalomnevek ($A_i = B_j$ sem fordulhat elő)
  - $R_1, \ldots, R_m$ különböző szerepnevek,
  - $S_1, \ldots, S_n$ is különböző szerepnevek (de $R_i = S_j$ előfordulhat),
  - $C_1, \ldots, C_m$ normálalakú kifejezések, de ha $R_i = S_j$ akkor $C_j \neq \bot$;
  - $k, l, m, n \geq 0$
- ha $k = l = m = n = 0$ (üres metszet) akkor a fogalom $\equiv \top$, de ez csak legkívül megengedett, A $\forall S_j.C_j$-beli C_j nem lehet üres metszet

A strukturális alárendeltségi algoritmus $\mathcal{AL}$ nyelvre

- A feladat: a $C \sqsubseteq D$ fogalomtartalmazás eldöntése, ahol C és D normálalakú $\mathcal{AL}$-fogalmak:
  1. Ha $C = \bot$, akkor kilép $C \sqsubseteq D$ eredménnyel.
  2. Ha $D = \bot$, akkor kilép $C \not\sqsubseteq D$ eredménnyel.
  3. Ha D-ben van olyan $\forall R.D'$ tag, amelyhez nincs C-ben egy $\forall R.C'$ tag, akkor kilép $C \not\sqsubseteq D$ eredménnyel.
  4. Ha D-ben van olyan nem $\forall R.D''$ alakú D' tag, amely nem szerepel C-ben, akkor kilép $C \not\sqsubseteq D$ eredménnyel.
  5. Egyébként, minden a D-beli $\forall R.D'$ tag és az ennek megfelelő C-beli $\forall R.C'$ tag (ilyen tag biztos van, a 3. pont miatt) esetén rekurzív módon megvizsgáljuk a $C' \sqsubseteq D'$ fogalomtartalmazást.
     - Ha ezek mindegyike igaz, akkor kilép $C \sqsubseteq D$ eredménnyel;
     - egyébként kilép $C \not\sqsubseteq D$ eredménnyel.
- Példák:
  - $A \sqcap \forall S.(B \sqcap \forall R.\bot) \sqcap \forall R.B \overset{?}{\sqsubseteq} A \sqcap \forall S.\forall R.\neg A \sqcap \forall R.(B \sqcap \neg A)$
  - $A \sqcap \forall S.(B \sqcap \forall R.\bot) \sqcap \forall R.(B \sqcap \neg A) \overset{?}{\sqsubseteq} A \sqcap \forall S.\forall R.\neg A \sqcap \forall R.B$

Tartalom

A tabló-algoritmusokról általában

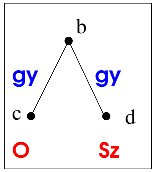
- Mindenféle logikákra (elsőrendű, modális, leíró) van tabló-algoritmus
- A leíró logikai tabló-algoritmus alapelvei:
  - Kielégíthetőség-vizsgálat konstruktív modellépítéssel
  - Negációs normálalak: negáció ($\neg$) csak atomi fogalmak előtt lehet
  - A modell építésekor következtetési szabályokat alkalmazunk
- Az épített modell
  - egy olyan gráf, amelynek élei és csomópontjai is címkézettek
  - a gráf csomópontjai alkotják az interpretáció alaphalmazát
  - a gráf éleit szerepekkel címkézzük (ezzel definiálva a szerepeket)
  - a gráf csomópontjait fogalomkifejezésekkel címkézzük, egy adott atomi fogalommal címkézett csomópontok definiálják a fogalmat
- Példa-feladat: Akinek van szép gyereke is és okos gyereke is, annak biztosan van-e olyan gyereke aki szép és okos is?

  $(\exists gy.O) \sqcap (\exists gy.Sz) \overset{?}{\sqsubseteq} \exists gy.(O \sqcap Sz)$
- Átfogalmazás: kielégíthető-e: $(\exists gy.O) \sqcap (\exists gy.Sz) \sqcap \neg(\exists gy.(O \sqcap Sz))$

  ($A \sqsubseteq B \Leftrightarrow A \sqcap \neg B$ *nem kielégíthető.*)

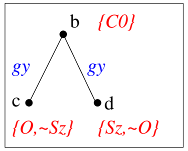
Leíró logikai tabló-algoritmus – $\mathcal{ALC}$ példa

- Kérdés: $(\exists gy.O) \sqcap (\exists gy.Sz) \overset{?}{\sqsubseteq} \exists gy.(O \sqcap Sz)$ (1)
- Átfogalmazás: kielégíthető-e $(\exists gy.O) \sqcap (\exists gy.Sz) \sqcap \neg(\exists gy.(O \sqcap Sz))$?
- Negált normálalak: $C_0 = (\exists gy.O) \sqcap (\exists gy.Sz) \sqcap \forall gy.(\neg O \sqcup \neg Sz)$
- Cél: olyan (véges) $\mathcal{I}$ interpretáció, amelyben $C_0^{\mathcal{I}} \neq \emptyset$. Tehát $\exists b$ egyed, amelyre $b \in (\exists gy.O)^{\mathcal{I}}, b \in (\exists gy.Sz)^{\mathcal{I}}$, és $b \in (\forall gy.(\neg O \sqcup \neg Sz))^{\mathcal{I}}$.
- $b \in (\exists gy.O)^{\mathcal{I}} \implies \exists c$ egyed, amelyre $\langle b, c \rangle \in gy^{\mathcal{I}}$ és $c \in O^{\mathcal{I}}$. Ugyanígy $b \in (\exists gy.Sz)^{\mathcal{I}} \implies \exists d.(\langle b, d \rangle \in gy^{\mathcal{I}}$ és $d \in Sz^{\mathcal{I}})$.
- Mivel $b \in (\forall gy.(\neg O \sqcup \neg Sz))^{\mathcal{I}}$; és c illetve d gy relációban állnak b-vel $\implies c \in (\neg O \sqcup \neg Sz)^{\mathcal{I}}$ illetve $d \in (\neg O \sqcup \neg Sz)^{\mathcal{I}}$.
- $c \in (\neg O \sqcup \neg Sz)^{\mathcal{I}}$ azt jelenti, hogy $c \in (\neg O)^{\mathcal{I}}$ vagy $c \in (\neg Sz)^{\mathcal{I}}$. Az első eset ellentmond $c \in O^{\mathcal{I}}$-nek. Tehát $c \in (\neg Sz)^{\mathcal{I}}$. Ugyanígy $d \in (\neg O \sqcup \neg Sz)^{\mathcal{I}}$ miatt $d \in (\neg O)^{\mathcal{I}}$.
- C_0-ra kaptunk egy modellt: $\Delta^{\mathcal{I}} = \{b, c, d\}$
  $gy^{\mathcal{I}} = \{\langle b, c \rangle, \langle b, d \rangle\}$;
  $O^{\mathcal{I}} = \{c\}$ és $Sz^{\mathcal{I}} = \{d\}$.
  Itt $b \in C_0^{\mathcal{I}}$, azaz (1) nem áll fenn.



Leíró logikai tabló-algoritmus – kiterjesztett ($\mathcal{ALCN}$) példa

- Kérdés: Akinek legfeljebb egy gyereke van, van szép gyereke is és van okos gyereke is, annak biztosan van-e olyan gyereke aki szép és okos is?
- LL-kérdés: $(\leqslant 1 gy) \sqcap (\exists gy.O) \sqcap (\exists gy.Sz) \overset{?}{\sqsubseteq} \exists gy.(O \sqcap Sz)$
- Átfogalmazás: kielégíthető-e $(\leqslant 1 gy) \sqcap (\exists gy.O) \sqcap (\exists gy.Sz) \sqcap \neg(\exists gy.(O \sqcap Sz))$
- Negált normálalak: $C_0 = (\leqslant 1 gy) \sqcap (\exists gy.O) \sqcap (\exists gy.Sz) \sqcap \forall gy.(\neg O \sqcup \neg Sz)$
- (1)-hez hasonlóan létrehozzuk az alábbi tablót:



- $(\leqslant 1 gy)(b)$, valamint $gy(b,c)$, $gy(b,d)$ miatt $c = d$ kell legyen, de az így kapott összevont egyed címkéjében két ütközés (ellentmondás) is van
- Ezzel „bebizonyítottuk", hogy C_0-nak nem lehet modellje, tehát a fenti kérdésre *igen* a válasz.

Az $\mathcal{ALCN}$ tabló-algoritmus menete

- Az algoritmus feladata: eldönteni, hogy egy adott C kielégíthető-e (egyelőre üres T-doboz felett) $\Longrightarrow$ keressünk egy modellt!
- Első lépésként C-t negációs normálalakra hozzuk, legyen ez C_0.
- Fő adatstruktúra: a modellt képviselő $\mathbf{T} = \langle V, E, \mathcal{L} \rangle$ *tabló* (irányított fa)
  - a fa csúcsai (V) az egyedek,
  - a fa élei (E) az egyedek között fennálló relációk
  - a csúcsokat és az éleket $\mathcal{L}$ címkékkel látja el:
    $v \in V$ esetén $\mathcal{L}(x) \subseteq sub(C_0)$,
    ahol $sub(D) = D$ részkifejezéseinek halmaza
    $\langle x, y \rangle \in E$ esetén $\mathcal{L}(\langle x, y \rangle)$ egy C-ben előforduló szerepnév
  - A kezdeti fa egy csúcsból áll: $\mathbf{T}_0 = \langle \{x_0\}, \emptyset, \mathcal{L} \rangle$ ahol $\mathcal{L}(x_0) = \{C_0\}$.
- Az algoritmus a fát transzformációs szabályok segítségével *bővíti*.
- Egyes szabályok nemdeterminisztikusak $\rightsquigarrow$ választási pont: visszalépés ellentmondás (ütközés) esetén, pl. A és $\neg A$ is $\in \mathcal{L}(x)$
- ha a fa *teljes* (azaz nem alkalmazható rá egy transzformációs szabály sem) és ellentmondásmentes $\Longrightarrow$ KILÉP: a C fogalom kielégíthető,
- Ha a teljes keresési téret bejártuk $\Longrightarrow$ KILÉP: C nem kielégíthető.

Az $\mathcal{ALCN}$ tabló-algoritmus menete (folyt.)

- Egyenlőségek kezelése
  - láttuk, hogy a $\leqslant$ kezeléséhez szükséges csomópontok összevonása
  - látjuk majd, hogy a $\geqslant$ kezeléséhez szükséges a csúcsok megkülönböztetése, ezért bevezetünk egy $\neq$ speciális szerepet:
    - $x \neq y$ csakkor teljesül az $\mathcal{I}$ interpretációban, ha $x^{\mathcal{I}} \neq y^{\mathcal{I}}$.
    - $\neq$ szimmetrikus, azaz ha $x \neq y$ igaz, akkor $y \neq x$ is teljesül.
  - A tabló adatstruktúrát kiterjesztjük: $\mathbf{T} = \langle V, E, \mathcal{L}, I \rangle$, ahol az I halmaz $x \neq y$ alakú állításokból áll ($x, y \in V$), I kezdetben üres.
  - Def.: x és y összevonható, ha $x \neq y \notin I$ és $y \neq x \notin I$
- A $\mathbf{T} = \langle V, E, \mathcal{L}, I \rangle$ tablónak megfelel egy $\mathcal{A}_{\mathbf{T}}$ A-doboz, melynek tartalma:
  - fogalmi állítások: $C(a)$ minden $a \in V$ és $C \in \mathcal{L}(a)$ esetén
  - szerepállítások: $R(a,b)$, minden $\langle a, b \rangle \in E$, $R = \mathcal{L}(\langle a, b \rangle)$ esetén
  - egyenlőtlenségek : $x \neq y$ minden $x \neq y \in I$ esetén (nem használjuk az UNA – egyedi név feltételezés – elvet)
- Def.: $\mathbf{T}$ *kielégíthető* ha $\mathcal{A}_{\mathbf{T}}$ kielégíthető ($\equiv \mathcal{A}_{\mathbf{T}}$ konzisztens)
- Az *R-követő* fogalma:

  ha egy a csúcsból vezet egy R-rel címkézett él a b csúcsba, akkor b-t az a csúcs *R-követőjének* (vagy csak követőjének) nevezzük

Negációs normálalak

- A negációs normálalakra való átalakítás szabályai

$$\neg\neg C \rightsquigarrow C$$

$$\neg(C \sqcap D) \rightsquigarrow \neg C \sqcup \neg D$$
$$\neg(C \sqcup D) \rightsquigarrow \neg C \sqcap \neg D$$

$$\neg(\exists R.C) \rightsquigarrow \forall R.\neg C$$
$$\neg(\forall R.C) \rightsquigarrow \exists R.\neg C$$

$$\neg(\leqslant nR) \rightsquigarrow (\geqslant n+1 R)$$
$$\neg(\geqslant 1 R) \rightsquigarrow \forall R.\bot$$
$$\neg(\geqslant nR) \rightsquigarrow (\leqslant mR) \; ahol \; m = n-1, \; n \geq 2$$

Az $\mathcal{ALCN}$ tabló transzformációs szabályai (1)

$\sqcap$-szabály

Feltétel:	$(C_1 \sqcap C_2) \in \mathcal{L}(x)$ és $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$
T′ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.

$\sqcup$-szabály

Feltétel:	$(C_1 \sqcup C_2) \in \mathcal{L}(x)$ és $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$.
T$_1$ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_1\}$.
T$_2$ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_2\}$.

$\forall$-szabály

Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$ és x-nek van olyan y R-követője, amelyre $C \notin \mathcal{L}(y)$.
T′ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{C\}$.

Az $\mathcal{ALCN}$ tabló transzformációk (2) – kiterjesztő szabályok

$\exists$-szabály

Feltétel:	$(\exists R.C) \in \mathcal{L}(x)$ és x-nek nincs olyan y R-követője, amelyre $C \in \mathcal{L}(y)$.
T′ *új állapot:*	$V' = V \cup \{y\}$ (y egy új csomópont),
	$E' = E \cup \{\langle x, y\rangle\}$, $\mathcal{L}'(\langle x, y\rangle) = R$, $\mathcal{L}'(y) = \{C\}$.

$\geqslant$-szabály

Feltétel:	$(\geqslant n R) \in \mathcal{L}(x)$ és x-nek nincs n olyan R-követője, amelyek között bármely kettő nem összevonható.
T′ *új állapot:*	$V' = V \cup \{y_1, \dots, y_n\}$ (y_i új csomópontok),
	$E' = E \cup \{\langle x, y_1\rangle, \dots, \langle x, y_n\rangle\}$,
	$\mathcal{L}'(\langle x, y_i\rangle) = R$, $\mathcal{L}'(y_i) = \emptyset$, minden $i = 1 \le i \le n$,
	$I' = I \cup \{y_i \neq y_j \mid 1 \le i < j \le n\}$.

Az $\mathcal{ALCN}$ tabló transzformációs szabályai (3)

$\leqslant$-szabály

Feltétel:	$(\leqslant n R) \in \mathcal{L}(x)$ és x-nek van $y_1, \dots, y_{n+1}$ R-követője, amelyek között van két összevonható.
	Minden olyan $(1 \le i < j \le n+1)$ esetén amelyre y_i és y_j összevonható:
T$_{ij}$ *új állapot:*	$V' = V \setminus \{y_j\}$, $\mathcal{L}'(y_i) = \mathcal{L}(y_i) \cup \mathcal{L}(y_j)$,
	$E' = E \setminus \{\langle x, y_j\rangle\} \setminus \{\langle y_j, u\rangle \mid \langle y_j, u\rangle \in E\} \cup \{\langle y_i, u\rangle \mid \langle y_j, u\rangle \in E\}$,
	$\mathcal{L}'(\langle y_i, u\rangle) = \mathcal{L}(\langle y_j, u\rangle)$, minden u esetén, melyre $\langle y_j, u\rangle \in E$,
	$I' = I[y_j \to y_i]$ (y_j minden előfordulását y_i-re cseréljük).

Az $\mathcal{ALCN}$ tabló-algoritmus — további részletek

- Kezdőállapot: $\mathbf{T}_0 = \langle \{x_0\}, \emptyset, \mathcal{L}, \emptyset \rangle$ ahol $\mathcal{L}(x_0) = \{C_0\}$ (C_0 = C NNF-ban)
- Ütközési feltételek
  - $\{\bot\} \subseteq \mathcal{L}(x)$ valamilyen $x \in V$ esetén;
  - $\{A, \neg A\} \subseteq \mathcal{L}(x)$ valamilyen $x \in V$, és A atomi fogalom esetén;
  - $\mathcal{L}(x)$ tartalmazza $(\leqslant nR)$-et, és x-nek van $y_1, \dots, y_{n+1}$ R-követője, amelyek között bármely kettő nem összevonható ($x \in V$).
- A tüzelési feltételek miatt egy helyen egy szabály csak egyszer alk.-ható
- A nemdeterminisztikus algoritmus általánosabb átfogalmazása:
  - Az algoritmus fák véges S halmazait transzformálja (a halmaz elemei az alternatív ágaknak felelnek meg)
  - Induláskor a halmaz egyetlen egycsucsú fát tartalmaz
  - Egy transzformációs szabályt egy $\mathbf{T} \in S$-re alkalmazunk, az új $S' = S \setminus \{T\} \cup S_{\mathbf{T}}$, ahol $S_{\mathbf{T}}$ a transzformáció által visszadott tabló-állapotok halmaza ($\{T'\}$, vagy $\{T_1, T_2\}$, vagy $\{T_{ij} | \cdots\}$).
  - Ha egy tablóban ütközés van, akkor azt elhagyjuk az S halmazból
  - A transzformációs folyamat akkor ér véget ha:
    - az S halmazban van egy teljes tabló-fa (kielégíthetőség)
    - az S halmaz üres (kielégíthetetlenség)

Példa az $\mathcal{ALCN}$ tabló-algoritmusra

- Vizsgáljuk az alábbi C_0 fogalom kielégíthetőségét (gy = gyereke, O = okos):

$$\begin{aligned} C_0 &= C_1 \sqcap C_2 \sqcap C_3 \sqcap C_4 \\ C_1 &= (\geqslant 2\,\text{gy}) \\ C_2 &= \exists \text{gy.O} \\ C_3 &= (\leqslant 2\,\text{gy}) \\ C_4 &= C_5 \sqcup C_6 \\ C_5 &= \forall \text{gy.}\neg\text{O} \\ C_6 &= \text{O} \end{aligned}$$

- A tabló-algoritmus által felépített modell: $\Delta^{\mathcal{I}} = \{b, c, d\}; \text{gy}^{\mathcal{I}} = \{\langle b, c\rangle, \langle b, d\rangle\}; \text{O}^{\mathcal{I}} = \{b, c\}$

Az $\mathcal{ALCN}$ tabló-algoritmus tulajdonságai

- Terminálás
  - A fában lefelé a címkék szerepmélysége határozottan csökken
  - A tabló-fa elágazási szélessége korlátos
  - A tabló-fa mint adatdoboz monoton bővül – ciklus nem lehetséges
- Teljesség: ha C kielégíthető, akkor az algoritmus ezt kimutatja
  - Azaz: ha az alg. nem-kielégíthetőséget jelez $\Rightarrow$ C nem kielégíthető.
  - Minden **T** tablónak megfelel egy $\mathcal{A}_{\mathbf{T}}$ adatdoboz.
  - A transzformációk megőrzik a kielégíthetőséget: ha a szabály egy **T** tablót egy $S_{\mathbf{T}}$ tablóhalmazba visz át, akkor $\mathcal{A}_{\mathbf{T}}$ kielégíthető $\Leftrightarrow$ ha van olyan $\mathbf{T}' \in S_{\mathbf{T}}$, hogy $\mathcal{A}_{\mathbf{T}'}$ kielégíthető.
- Helyesség: ha az alg. kielégíthetőséget jelez, akkor C kielégíthető
  - Építsünk egy $\mathcal{I}$ modellt, amelyben C nem üres (modellkonstrukció): Az $\mathcal{A} = \mathcal{A}_{\mathbf{T}}$ adatdobozhoz (ahol **T** teljes és ütközésmentes)
    - felépítjük az ún. *természetes* interpretációt
    - megmutatjuk, hogy ez az interpretáció $\mathcal{A}$ modellje
    - mivel $C_0(b) \in \mathcal{A}$ ezért $C \equiv C_0$ kielégíthető (C a vizsgált fogalom, C_0 a normálalakja)

Adatdobozok természetes interpretációja, önmegvalósítás

- $\mathcal{A}$ A-dobozhoz definiáljunk egy $\mathcal{I} = \mathcal{I}^{nat}(\mathcal{A})$ *természetes* interpretációt:

$$\begin{aligned} \Delta^{\mathcal{I}} &= \{a \mid C(a) \in \mathcal{A}, \text{ vagy } R(a,x) \in \mathcal{A}, \text{ vagy } R(y,a) \in \mathcal{A}\} \\ a^{\mathcal{I}} &= a, \text{ minden } a \in \Delta^{\mathcal{I}} \text{ egyednév esetén} \\ A^{\mathcal{I}} &= \left\{a \mid a \in \Delta^{\mathcal{I}}, A(a) \in \mathcal{A}\right\}, \text{ minden } \mathcal{A}\text{-beli } A \text{ fogalomnév esetén} \\ R^{\mathcal{I}} &= \left\{\langle a, b\rangle \mid a, b \in \Delta^{\mathcal{I}}, R(a,b) \in \mathcal{A}\right\}, \text{ minden } \mathcal{A}\text{-beli } R \text{ esetén} \end{aligned}$$

- Egy $\mathcal{A}$ adatdobozt *önmegvalósító*nak mondunk ha $\mathcal{I}^{nat}(\mathcal{A}) \models \mathcal{A}$
- Állítás: Ha **T** teljes és ütközésmentes, akkor $\mathcal{A}_{\mathbf{T}}$ önmegvalósító:
- Megmutatjuk, hogy $\mathcal{I} = \mathcal{I}^{nat}(\mathcal{A}_{\mathbf{T}})$ kielégít minden $\mathcal{A}_{\mathbf{T}}$-beli állítást.
  - Atomi fogalom- illetve szerepállítások esete. Ezek az $\mathcal{I}^{nat}$ fenti definíciója szerint nyilvánvalóan igazak $\mathcal{I}$-ben.
  - A $(\neg A)(x)$ alakú fogalomállítások. **T** ütközésmentes $\Rightarrow A(x) \notin \mathcal{A}_{\mathbf{T}}$, így a def. szerint $x \notin A^{\mathcal{I}}$, tehát $(\neg A)(x)$ igaz $\mathcal{I}$-ben.
  - A $C(x)$ alakú fogalomállítások, ahol C nem atomi fogalom és nem is negált atomi fogalom. Egy ilyen állítás esetén a kifejezés szerkezetére vonatkozó indukcióval bizonyítjuk az igazságát, kihasználva **T** teljességét.

A természetes interpretáció modellje az $\mathcal{A}_{\mathbf{T}}$ A-doboznak

- Áll.: ha **T** teljes és ütközésmentes, akkor $\mathcal{A}_{\mathbf{T}}$ önmegvalósító
- Bizonyítandó még, hogy $C(x) \in \mathcal{A}_{\mathbf{T}} \Rightarrow \mathcal{I}^{nat} \models C(x)$, (1) ahol C lehet $D \sqcap E$, $D \sqcup E$, $\exists R.D$, $\forall R.D$, $(\geq n\,R)$, $(\leq n\,R)$ alakú
- Indukciós feltevés: ha C' részfogalma C-nek akkor (1) igaz C'-re
- T.f.h. $C = \exists R.D$. Mivel $C(x) \in \mathcal{A}_{\mathbf{T}} \Rightarrow \exists R.D \in \mathcal{L}(x)$
- **T** teljes, így x-re a $\exists$-szabály nem alkalmazható, ez csak úgy lehet, hogy $\exists y.(\mathcal{L}(\langle x, y\rangle) = R, D \in \mathcal{L}(y))$
- $D \in \mathcal{L}(y) \Rightarrow D(y) \in \mathcal{A}_{\mathbf{T}} \Rightarrow$ (indukcióval) $\mathcal{I}^{nat} \models D(y)$
- Mivel $\mathcal{I}^{nat} \models D(y)$ és $\mathcal{I}^{nat} \models R(x,y)$, ezért az $\mathcal{ALCN}$ szemantika szerint $\mathcal{I}^{nat} \models (\exists R.D)(x)$, q.e.d.
- A többi konstruktor esete hasonlóan kezelhető, minden esetben kihasználjuk a tabló teljességét!

T-dobozok kezelése

- Minden T-doboz felfogható, mint általános tartalmazási axiómák ($C \sqsubseteq D$) halmaza ($C \equiv D$ helyettesíthető: $\{C \sqsubseteq D, D \sqsubseteq C\}$-vel).
- $\mathcal{T} = \{C_1 \sqsubseteq D_1, C_2 \sqsubseteq D_2, \ldots, C_n \sqsubseteq D_n\} \equiv \{\top \sqsubseteq C_{\mathcal{T}}\}$, ahol:

$$C_{\mathcal{T}} = (\neg C_1 \sqcup D_1) \sqcap (\neg C_2 \sqcup D_2) \sqcap \cdots \sqcap (\neg C_n \sqcup D_n).$$

- $C_{\mathcal{T}}$-be tehát egy modell minden elemének kötelessége beletartozni.
- Ehhez minden csomópont címkéjéhez hozzá kell vennünk $C_{\mathcal{T}}$-t.
- Új kezdőállapot: $\mathbf{T}_0 = \langle \{x_0\}, \emptyset, \mathcal{L}, \emptyset \rangle$ ahol $\mathcal{L}(x_0) = \{C_0, C_{\mathcal{T}}\}$ (A továbbiakban is színezés és aláhúzás jelezi a módosításokat.)
- Minden új csúcs létrehozásakor a cimkébe bekerül $C_{\mathcal{T}}$ $\Rightarrow \infty$ ciklus veszélye!
  - Példa: vizsgáljuk a B fogalom kielégíthetőségét $\{\top \sqsubseteq \exists$gy.O$\}$ felett!
  - Minden csomópont címkéjébe belekerül a $\exists$gy.O fogalom
  - emiatt a $\exists$-szabály alkalmazása végtelen gyermek-láncot generál.
- A végtelen ciklus kiküszöbölésére a *blokkolás* módszerét használjuk

Blokkolás

- Definíció: egy y csomópontot blokkol az x csomópont, ha x a fában y felett van, és $\mathcal{L}(y) \subseteq \mathcal{L}(x)$ (*részhalmaz blokkolás*).
- Ha y blokkolt $\Rightarrow$ y-ra nem alkalmazzuk a kiterjesztő ($\exists$ és $\geqslant$) szabályokat.
- Így megoldódik a terminálási probléma, de:
  - Kérdés: ha a tablóban x blokkolja y-t, hogyan építsünk modellt?
  - Válasz ($\mathcal{ALC}$ esetén): azonosítsuk az y és x csomópontokat, mert: x címkéjében y összes címkefogalma szerepel, így x minden olyan C-ben benne van, amelyben y-nak benne kell lennie
- Pl. kielégíthető-e Szep $\sqcap$ Okos a $\{\top \sqsubseteq \exists$gyermeke.Okos$\}$ T-doboz felett?
  - A példa tablója:

$$\begin{array}{rcl} & x \circ & \{\text{Szep}, \text{Okos}, \exists\text{gyermeke.Okos}\} \\ \texttt{gyermeke} & | & \\ & y \circ & \{\text{Okos}, \exists\text{gyermeke.Okos}\} \end{array}$$

  - x blokkolja y-t
  - a modell: $\Delta^{\mathcal{I}} = \{x\}; \text{Szep}^{\mathcal{I}} = \{x\}; \text{Okos}^{\mathcal{I}} = \{x\}; \text{gyermeke}^{\mathcal{I}} = \{\langle x, x \rangle\}$

Tabló-algoritmus blokkolással – részletek

- Egy $\mathbf{T} = \langle V, E, \mathcal{L}, I \rangle$ tabló-gráfra vonatkozó definíciók:
  - Egy y csomópont az x *R-követője*, ha $\langle x, y \rangle \in V$ és $R = \mathcal{L}(\langle x, y \rangle)$. Ebben az esetben azt is mondjuk, hogy y az x követője.
  - Egy x csomópont az y csúcs *megelőzője*, ha y az x-nek követője.
  - Egy x csomópont a z csúcs *őse*, ha x a z-nek megelőzője, vagy ha z-nek van olyan y őse, amelynek megelőzője x. Azaz az „őse" kapcsolat a „megelőzője" kapcsolat tranzitív lezártja.
  - Egy z csúcs az x csúcs *leszármazottja*, ha z-nek őse x.
- Egy y csúcsot részhalmaz-blokkolja egy x őse, ha $\mathcal{L}(y) \subseteq \mathcal{L}(x)$.
- Statikus a blokkolás, ha x egyszer blokkolódik, mindig blokkolt marad
- A statikus blokkolás érdekében megszorítjuk a szabályok alk.-i sorrendjét: a kiterjesztő szabályok csak ún. *stabil* csúcsra alkalmazhatók
- Egy x csúcs *stabil*, ha őseire semmilyen szabály sem alkalmazható, és x-re is csak kiterjesztő ($\exists$- és/vagy $\geqslant$-) szabályok alkalmazhatóak.
- Másszóval: a kiterjesztő szabályokat utolsóként alkalmazzuk, ha az egész tabló-gráfban semmilyen más szabály nem alkalmazható

A T-dobozos tabló-algoritmus megváltozott szabályai

$\exists$-szabály	
Feltétel:	$(\exists R.C) \in \mathcal{L}(x)$, nincs olyan y, amelyre $\mathcal{L}(\langle x, y \rangle) = R$ és $C \in \mathcal{L}(y)$, továbbá x stabil és nem blokkolt
T′ *új állapot:*	$V' = V \cup \{y\}$, $E' = E \cup \{\langle x, y \rangle\}$, $\mathcal{L}'(\langle x, y \rangle) = R$, $\mathcal{L}'(y) = \{C, C_{\mathcal{T}}\}$.
$\geqslant$-szabály	
Feltétel:	$(\geqslant n\, R) \in \mathcal{L}(x)$, x-nek nincs n olyan R-követője, amelyek között bármely kettő nem összevonható, továbbá x stabil és nem blokkolt
T′ *új állapot:*	$V' = V \cup \{y_1, \ldots, y_n\}$, $E' = E \cup \{\langle x, y_1 \rangle, \ldots, \langle x, y_n \rangle\}$, $\mathcal{L}'(\langle x, y_i \rangle) = R$, $\mathcal{L}'(y_i) = \{C_{\mathcal{T}}\}$, minden $i = 1 \leq i \leq n$, $I' = I \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\}$.

A T-dobozos $\mathcal{ALCN}$ tabló-algoritmus tulajdonságai

- Terminálás: A fában csak véges sok fogalomkifejezés fordulhat elő, ezért minden ág előbb-utóbb blokkolódik.
- Teljesség: változatlan érvelés
- Helyesség (legyen $\mathbf{T}$ teljes ütközésmentes tabló):
  Az $\mathcal{A}_\mathbf{T}$ adatdobozt bővítsük az alábbi $\mathcal{A}'_\mathbf{T}$ A-dobozzá:

$$\begin{aligned}\mathcal{A}'_\mathbf{T} \quad = \quad & \mathcal{A}_\mathbf{T} \cup \\ & \left\{R(y,z) \mid y\text{-t } x \text{ blokkolja, } (\exists R.C)(y) \in \mathcal{A}_\mathbf{T} \text{ és } z = succ^\mathbf{T}_{\exists R.C}(x)\right\} \cup \\ & \left\{R(y,z) \mid y\text{-t } x \text{ blokkolja, } (\geqslant n\, R)(y) \in \mathcal{A}_\mathbf{T} \text{ és } z \in succSet^\mathbf{T}_{(\geqslant n\, R)}(x)\right\}\end{aligned}$$

- A fenti formulákban használt jelölések:
  - Ha $\exists R.C \in \mathcal{L}(x)$, akkor jelöljük $z = succ^\mathbf{T}_{\exists R.C}(x)$-szel x-nek egy olyan R-követőjét, amelyre $C \in \mathcal{L}(z)$ (ilyen van, $\mathbf{T}$ teljessége miatt).
  - Ha $(\geqslant n\, R) \in \mathcal{L}(x)$, akkor legyen $succSet^\mathbf{T}_{(\geqslant n\, R)}(x)$ egy olyan $\{z_1, \ldots, z_n\}$ csúcshalmaz, ahol minden z_i x-nek R-követője, és $z_i \neq z_j, 1 \leq i < j \leq n$ (ilyen van, $\mathbf{T}$ teljessége miatt).
- Állítás: $\mathcal{A}'_\mathbf{T}$ önmegvalósító.

Adatdobozok kezelése

- **Kérdés**: egy $\mathcal{A}$ adatdoboz konzisztens-e egy $\mathcal{T}$ T-doboz felett?
- **Kezdeti tabló**: $\mathcal{A}$-ból felépítünk egy kezdeti $\mathbf{T}_0$ tabló-állapotot (lényegében a $\mathbf{T} \to \mathcal{A}_\mathbf{T}$ transzformáció megfordításával):
  - a tabló csúcsai az $\mathcal{A}$-ban előforduló $\{a_1, \ldots, a_n\}$ egyednevek;
  - az a_i és a_j csúcsok között pontosan akkor megy egy R címkéjű él, ha $\mathcal{A}$-ban szerepel egy $R(a_i, a_j)$ szerepállítás;
  - egy a_i csomópont címkéje tartalmazza a $C_\mathcal{T}$ belsősítést, és azon D fogalmakat, amelyekre $\mathcal{A}$-ban van egy $D(a_i)$ állítás;
  - a tabló-állapot egyenlőtlenség-rendszerébe az összes egyednév-párt felvesszük, az egyedinév-kikötés biztosítására (ha ez szükséges).
- **Előfeldolgozás**: A $\mathbf{T}_0$ kezdeti tabló-állapotra és a $\mathcal{T}$ T-dobozra alkalmazzuk a tabló-algoritmust, a következő két megkötéssel:
  - Kiterjesztő szabályt csak az eredeti adatdoboz a_i csomópontjaira szabad alkalmazni, azok követőire már nem.
  - Ha egy a_i csomópontot egy, az eredeti adatdobozban nem szereplő x csomóponttal vonunk össze, akkor a_i-t tartjuk meg, azaz x címkéit vesszük hozzá a_i címkehalmazához, és nem fordítva.

Adatdobozok kezelése (2)

- A nemdeterminisztikus szabályok miatt az előfeldolgozás eredménye több tabló-állapot is lehet, ezek közül hagyjuk el az ütközést tartalmazókat és jelöljük a fennmaradó állapotok halmazát $S_\mathcal{A}$-val.
- **Kielégíthetőség-vizsgálat**: Az előfeldolgozási fázisban kapott $\mathbf{T} = \langle V, E, \mathcal{L}, I \rangle \in S_\mathcal{A}$ tabló-állapotokat sorra vesszük, és ezeket mindaddig vizsgáljuk, míg valamelyiket kielégíthetőnek találjuk a következő értelemben:
  - Az $\mathcal{A}$ adatdoboz minden egyes a_i egyednevéhez felépítjük azt a C_i fogalomkifejezést amely az a_i csomópont $\mathbf{T}$-beli címkehalmazában levő fogalmak metszete: $C_i = \sqcap_{D \in \mathcal{L}(a_i)} D$
  - A tabló-algoritmus n-szeri alkalmazásával eldöntjük, hogy a C_i fogalmak mindegyike kielégíthető-e a $\mathcal{T}$ terminológiai doboz felett. Ha ez teljesül, akkor a tabló-állapotot kielégíthetőnek mondjuk, az algoritmus véget ér, és az „$\mathcal{A}$ konzisztens $\mathcal{T}$ felett" választ adja.
  - Egyébként folytatjuk a soron következő $S_\mathcal{A}$-beli tablóval.

  Ha az $S_\mathcal{A}$-beli tabló-állapotok egyikét sem találtuk kielégíthetőnek, akkor az algoritmus az „$\mathcal{A}$ nem konzisztens $\mathcal{T}$ felett" válasszal ér véget.

Tartalom

A tabló-algoritmus áttekintése

- Kérdés: Egy NNF alakú C fogalom kielégíthető-e, egy (esetleg üres) $\mathcal{T}$ T-doboz felett.
- Az algoritmus **transzformációs szabályokat** alkalmaz tabló-(állapoto)kra
- Egy $\mathbf{T} = \langle V, E, \mathcal{L}, I \rangle$ tabló-állapot: véges, irányított, címkézett gráf (fa) plusz egy csúcsokra vonatkozó egyenlőtlenség-rendszer.
  - Egy $\mathbf{T}$ tabló-állapothoz hozzárendelhető egy $\mathcal{A}_\mathbf{T}$ adatdoboz:
    - a gráf csomópontjai $\rightarrow$ egyednevek
    - egy csomópont címkehalmaza $\rightarrow$ fogalmi állítások
    - egy címkézett él $\rightarrow$ egy szerepállítás (később több is lehet majd)
  - Transzf. szabály: tüzelési feltétel $\Rightarrow$ $\mathbf{T}$-hez $\mathcal{S}_\mathbf{T}$ tabló-halmazt rendel
  - Többelemű $\mathcal{S}_\mathbf{T}$ állapothalmaz: választási pont (nemdet. keresés)
- Kielégíthetőség megőrzése: $\mathcal{A}_\mathbf{T}$ kielégíthető $\Leftrightarrow$ ha van $\mathbf{T}' \in \mathcal{S}_\mathbf{T}$ tabló, hogy $\mathcal{A}_{\mathbf{T}'}$ kielégíthető.
- A kiindulási tabló: $\mathbf{T}_0(C) = \langle \{x_0\}, \emptyset, (\mathcal{L}(x_0) \mapsto \{C\}), \emptyset \rangle$ $\mathcal{A}_{\mathbf{T}_0}$ kielégíthető $\Leftrightarrow$ ha C kielégíthető.
- Ütközés: olyan ellentmondás $\mathbf{T}$-ben, ami miatt $\mathcal{A}_\mathbf{T}$ nem lehet kielégíthető.
  - Pl. egy csúcs címkéje tartalmazza $\bot$-t, vagy a $\neg A$ és A fogalmakat

A tabló-algoritmus áttekintése (2)

- „Sikeres” lefutás: ha egy ütközésmentes és teljes állapotba jutunk
  - $\mathbf{T}$ teljes, ha semmilyen transzf. szabály sem alkalmazható rá.
  - A „sikeres” eredmény jogos, mert egy teljes és ütközésmentes tablóhoz építhető $\mathcal{T}$-nek olyan modellje, hogy C nem üres — **modellkonstrukció**
- „Sikertelen” lefutás: ha a keresési tér minden ágán ütközést észlelünk.
  - A „sikertelen” eredmény jogos, mert a transzf.-k megőrzik a kielégíthetőséget, és az ütközés nyilvánvaló kielégíthetetlenség
- A tabló-algoritmus véget ér, mivel
  - a tabló-gráf mérete C és $\mathcal{T}$ jellemzőivel felülről korlátozható,
  - a transzformációs szabály monotonok
  - a végtelen ciklust meggátolja a **blokkolás** (többfajta, a modellkonstrukciótól függően)
- A $\mathcal{SHIQ}$ tabló-algoritmus bemutatásának menetrendje
  - sorra vesszük az $\mathcal{S}$, $\mathcal{SH}$, $\mathcal{SHI}$, $\mathcal{SHIF}$ és $\mathcal{SHIQ}$ tabló-algoritmusokat
  - a fókuszban: transzf. szabályok, blokkolásfajták, modellkonstrukció

A tabló-algoritmus rekurzív változata (mélységi keresés)

- C kielégíthetőségének eldöntése: a *kielégíthető_tabló*$(\mathbf{T}_0(C))$ hívással

procedure *kielégíthető_tabló*$(\mathbf{T})$:

1. Ha $\mathbf{T}$-ben ütközés van $\Rightarrow$ „hamis” (kilépés „hamis” eredménnyel).
2. Egyébként, ha $\mathbf{T}$ teljes, akkor $\Rightarrow$ „igaz”.
3. Egyébként van olyan x csúcs, melyre egy szabály alkalmazható $\mathbf{T}$-ben. Tetszőleges módon válasszunk (*) egy ilyen csúcs-szabály párt, hajtsuk végre a szabályt, állítsuk elő az $\mathcal{S}_\mathbf{T}$ állapothalmazt, és legyen $S := \mathcal{S}_\mathbf{T}$!
4. Válasszunk (*) egy $\mathbf{T}' \in S$ állapotot, és legyen $S := S \setminus \{\mathbf{T}'\}$!
5. Rekurzívan futtassuk a *kielégíthető_tabló*$(\mathbf{T}')$ hívást!
6. Ha az eredmény „igaz”, akkor $\Rightarrow$ „igaz”.
7. Egyébként, ha $S = \emptyset$, akkor $\Rightarrow$ „hamis”.
8. Egyébként folytassuk az eljárást a 4. lépéssel!

(*) A választás „don’t care” jellegű, csak egy lehetőséget kell választani.

A tabló-algoritmus iteratív változata (tetsz. keresés)

- C kielégíthetőségének eldöntése: a *kielégíthető_fogalom*(C) hívással

procedure *kielégíthető_fogalom*(C):

1. Legyen $S := \{\mathbf{T}_0(C)\}$!
2. Ha az S halmaz minden eleme ütközést tartalmaz $\Rightarrow$ „hamis”.
3. Egyébként, ha S-ben van teljes ütközésmentes tabló $\Rightarrow$ „igaz”.
4. Egyébként S-ben van nem-teljes ütközésmentes tabló. Válasszunk (*) egy ilyen $\mathbf{T} \in S$ állapotot, majd válasszunk (*) $\mathbf{T}$-ben egy olyan csúcsot, amelyre alkalmazható egy szabály, hajtsuk végre ezt a szabályt, és állítsuk elő az $\mathcal{S}_\mathbf{T}$ állapothalmazt!
5. Az S halmazban cseréljük le $\mathbf{T}$-t $\mathcal{S}_\mathbf{T}$-re: $S := S \setminus \{\mathbf{T}\} \cup \mathcal{S}_\mathbf{T}$!
6. Folytassuk a 2. lépésnél!

(*) A választás „don’t care” jellegű, csak egy lehetőséget kell választani.

Az $\mathcal{S}$ nyelv

- $\mathcal{S} \equiv \mathcal{ALC}_{\mathcal{R}^+}$, azaz az $\mathcal{ALC}$ nyelv kiegészítve tranzitív szerepekkel:
  - A tranzitivitási axióma alakja: Trans(R) $\equiv$ az R szerep tranzitív.
  - Példa: Trans(`őse`) ($\equiv$ az ősök ősei is ősök)
- Induljunk ki a (blokkolásos) $\mathcal{ALC}$ tabló-algoritmusból
  - A tranzitivitás *explicit* kezelése: ha Trans(R), x R-követője y, y R-követője z, $\Rightarrow$ egy új $x \to z$ él szükséges, R címkével
  - Ez nem hatékony, a tranzitivitást *implicit* módon kezeljük
  - Mire is hatna az új él? Lényegében csak a $\forall$-szabályra.
- A $\forall$-szabály (ism.): $\forall R.C \in \mathcal{L}(x) \longrightarrow x$ $\forall R$-követője kap egy C címkét:

$\forall$-szabály

Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$, x-nek $\exists y$ R-követője, hogy $C \notin \mathcal{L}(y)$.
T′ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{C\}$.

- Az új él hatása kiváltható egy hasonló $\forall_+$-szabállyal: ha R tranzitív és $(\forall R.C) \in \mathcal{L}(x) \longrightarrow x$ minden R-követője megkapja a $(\forall R.C)$ címkét is.

A $\forall_+$ szabály a tranzitivitás kezelésére

$\forall_+$-szabály

Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$, Trans$(R) \in \mathcal{T}$ és x-nek van olyan y R-követője, hogy $\forall R.C \notin \mathcal{L}(y)$.
T′ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{\forall R.C\}$.

- A $\forall_+$ szabály miatt a cimkék ismétlődhetnek $\Longrightarrow$ blokkolás szükséges
- 1. példa: $\exists$lsz.$\top \sqcap \forall$lsz.$\exists$lsz.$\top$ kielégíthető-e $\{$Trans(lsz)$\}$ felett? (Itt lsz = leszármazottja)
- 2. példa: O $\sqcap$ $\exists$lsz.Sz $\sqcap$ $\forall$lsz.$\exists$lsz.Sz kielégíthető-e Trans(lsz) felett?

```
          x o   {O, ∃lsz.Sz, ∀lsz.(∃lsz.Sz)}
lsz         |
          y o   {Sz, ∃lsz.Sz, ∀lsz.(∃lsz.Sz)}
lsz         |
          z o   {Sz, ∃lsz.Sz, ∀lsz.(∃lsz.Sz)}        y blokkolja z-t
```

Modellkonstrukció: a blokkolt z pontot vonjuk össze a blokkoló y-nal:
$\Delta^{\mathcal{I}} = \{x, y\}$; $\mathrm{O}^{\mathcal{I}} = \{x\}$; $\mathrm{Sz}^{\mathcal{I}} = \{y\}$; $\mathrm{lsz}^{\mathcal{I}} = \{\langle x, y \rangle, \langle y, y \rangle\}$

Az $\mathcal{S}$ nyelv – modellkonstrukció

- Legyen $\mathbf{T} = \langle V, E, \mathcal{L}(,I) \rangle$ egy teljes és ütközésmentes $\mathcal{S}$-tabló-állapot.
- A $\mathbf{T}$-nek megfeleltetett adatdoboz (ism.):

$$\mathcal{A}_{\mathbf{T}} = \{D(x) \mid x \in V, D \in \mathcal{L}(x)\} \cup \{R(x,y) \mid \langle x, y \rangle \in E, R = \mathcal{L}(\langle x, y \rangle)\}$$

- Vonjunk össze minden blokkolt csúcsot a blokkoló csúccsal:

$$\begin{aligned} \mathit{IdBlock}_{\mathbf{T}}\, \mathcal{A} &= \{C(id_a) \mid C(a) \in \mathcal{A}\} \cup \\ &\quad \{R(id_a, id_b) \mid R(a,b) \in \mathcal{A}\}\text{, ahol} \end{aligned}$$

$$id_x = \begin{cases} x & \text{ha } x \text{ nem blokkolt a } \mathbf{T} \text{ tablóban} \\ y & \text{ha az } x \text{ csúcsot } y \text{ blokkolja a } \mathbf{T} \text{ tablóban} \end{cases}$$

- A tablóban a tranzitivitás implicit, „rakjuk bele” az adatdobozba

$$\begin{aligned} \mathit{ClosT}_{\mathcal{T}}\, \mathcal{A} = \mathcal{A} \cup \{\, R(a_1, a_n) \mid & \,\mathbf{Trans}(R) \in \mathcal{T}, n > 2\text{, és} \\ & \{R(a_1, a_2), \ldots, R(a_{n-1}, a_n)\} \subseteq \mathcal{A} \,\} \end{aligned}$$

- A lezárásokkal *önmegvalósító* A-dobozt kapunk, így a keresett modell:

$$\mathcal{I}_{\mathbf{T}} = \mathcal{I}^{nat}(\mathit{ClosT}_{\mathcal{T}}\; \mathit{IdBlock}_{\mathbf{T}}\; \mathcal{A}_{\mathbf{T}})$$

Szerephierarchiák – a $\mathcal{H}$ nyelvkiterjesztés

- Szerephierarchia ($\mathcal{H}$): a T-dobozban lehetnek $R \sqsubseteq S$ alakú szerepállítások.
- Az $\mathcal{SH}$ nyelvre és az annál nagyobb kifejezőerejű nyelvekere alkalmazható a belsősítés módszere: a fogalmi állítások kiküszöbölhetők a T-dobozból.
- Fontos! Mostantól kezdve feltételezzük, hogy a T-doboz kizárólag szerepaxiómákat ($R \sqsubseteq S$, $\mathbf{Trans}(R)$) tartalmaz.
- Hogyan kezeljük a szerephierarchiát a transzformációs szabályokban?
  - Példa: $C = \exists$gyereke.$\neg$Boldog $\sqcap$ $\forall$rokona.Boldog kielégíthető-e, a $\{$gyereke $\sqsubseteq$ rokona$\}$ T-doboz felett?

```
b •  {C,  ...,  ∀rokona.Boldog}
  |
  |  gyereke
  |
c •  {¬Boldog}
```

C nyilván nem kielégíthető, de ehhez a $\forall$-szabályt módosítani kell.

Szerephierarchiák (2)

- Szükséges a szereptartalmazási axiómák „lezárása”, pl.
  fia $\sqsubseteq$ gyereke és gyereke $\sqsubseteq$ leszármazottja $\Longrightarrow$ fia $\sqsubseteq$ leszármazottja
- Legyen $\mathcal{T}$ egy csak szerepax.-kat tartalmazó T-doboz és C egy fogalom. $\mathcal{T}$ tranzitív-reflexív lezárása az az $\mathcal{R}_{\mathcal{T}(,C)}$ legszűkebb halmaz, melyre
  - Minden $T \in \mathcal{T}$ esetén $T \in \mathcal{R}_\mathcal{T}$.
  - Ha az R szerep $\mathcal{T}$-ben vagy C-ben előfordul, akkor $R \sqsubseteq R \in \mathcal{R}_\mathcal{T}$
  - Ha $R \sqsubseteq R' \in \mathcal{R}_\mathcal{T}$, és $R' \sqsubseteq R'' \in \mathcal{R}_\mathcal{T}$, akkor $R \sqsubseteq R'' \in \mathcal{R}_\mathcal{T}$.
- Példa: ha $\mathcal{T} = \{\text{fia} \sqsubseteq \text{gy}, \text{gy} \sqsubseteq \text{lsz}, \textbf{Trans}(\text{lsz})\}$, $C = \exists\text{huga}.\top$ akkor
  $\mathcal{R}_{\mathcal{T},C} = \{$fia $\sqsubseteq$ gy, gy $\sqsubseteq$ lsz, ($\sqsubseteq$ tranzitivitása miatt:) fia $\sqsubseteq$ lsz,
  (reflex. miatt:) fia $\sqsubseteq$ fia, gy $\sqsubseteq$ gy, lsz $\sqsubseteq$ lsz, huga $\sqsubseteq$ huga,
  Trans(lsz) $\}$
- Módosítjuk az R-követő fogalmát (egy C fogalom $\mathcal{T}$ feletti kielégíthetőségét vizsgáló $\mathbf{T} = \langle V, E, \mathcal{L}, I \rangle$ tablóban):
  - Egy y csomópont az x *R-követője*, ha
    van olyan $S \sqsubseteq R \in \mathcal{R}_\mathcal{T}$, és $y \in V$, hogy $\langle x, y \rangle \in E$ és $S = \mathcal{L}(\langle x, y \rangle)$.
  - $\mathcal{R}_\mathcal{T}$ reflexivitása miatt ez konzervatív kiterjesztés (azaz ha nincs $R \sqsubseteq S$ alakú axióma, akkor a definíció ekvivalens a korábbival)

Szerephierarchiák: módosult transzformációs szabályok

- A $\forall$-szabály: annyi a változás, hogy az új „követője” kapcsolatot használja
- A $\forall_+$-szabály: az új „követője” kapcsolat használata nem elegendő, példa:
  - x címkéje: $\forall$rokona.Szép
  - x-nek van egy y gyereke-követője
  - gyereke $\sqsubseteq$ lsz, lsz $\sqsubseteq$ rokona, Trans(lsz)
- Mivel a rokona szerep nem tranzitív, a régi $\forall_+$-szabály nem tüzel
  - De $\forall$rokona.Szép $\sqsubseteq$ $\forall$lsz.Szép, és az lsz szerep tranzitív
  - Emiatt y címkéjébe fel kell venni a $\forall$lsz.Szép fogalmat
- A módosult szabály:

$\forall_+$-**szabály**	
Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$, van olyan S, hogy Trans$(S) \in \mathcal{R}_\mathcal{T}$ és $S \sqsubseteq R \in \mathcal{R}_\mathcal{T}$, továbbá x-nek van egy y S-követője, amelyre $(\forall S.C) \notin \mathcal{L}(y)$.
$\mathbf{T}'$ ***új állapot:***	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{\forall S.C\}$.

$\mathcal{SH}$ nyelv – modellkonstrukció

- Egy $\mathcal{A}$ adatdoboz lezárása egy $\mathcal{R}_\mathcal{T}$ szerephierarchia szerint (vegyük hozzá az adatdobozhoz a belőle és a szerephierarchiából együttesen következő állításokat):

$$ClosH_\mathcal{T}\ \mathcal{A} \quad = \quad \mathcal{A} \cup \{R(a, b) \mid (S \sqsubseteq R) \in \mathcal{R}_\mathcal{T}, S(a, b) \in \mathcal{A}\}$$

- Az adatdoboz lezárása a szerephierarchia és a tranzitivitás együttes figyelembevételével:

$$ClosHT_\mathcal{T}\ \mathcal{A} \quad = \quad ClosH_\mathcal{T}\ ClosT_\mathcal{T}\ ClosH_\mathcal{T}\ \mathcal{A}$$

  - A $ClosH_\mathcal{T}$ másodszori alkalmazása: tranzitív szerepet tartalmazó nem-tranzitív szerepek miatt szükséges.
- A $\mathbf{T}$ tablóhoz tartozó modell:

$$\mathcal{I}_\mathbf{T} \quad = \quad \mathcal{I}^{nat}(ClosHT_\mathcal{T}\ IdBlock_\mathbf{T}\ \mathcal{A}_\mathbf{T})$$

Inverz szerepek: az $\mathcal{I}$ nyelvkiterjesztés

- A fogalmak most már alulról felfelé is terjedhetnek a tablóban
- 1. példa: $C = \exists\text{gy}.\forall\text{gy}^-.\text{Boldog}$, építsünk tablót a $C \sqcap \neg\text{Boldog}$-hoz:

```
b •  {C, ¬Boldog}
  |
  |  gy
c •  {∀gy⁻.Boldog}
```

  A $\forall$-szabály `c`-re alkalmazva a *fölötte* levő `b` csúcs címkéjébe kell tegye a Boldog fogalmat!
- 2. példa $D = \text{Boldog} \sqcap \exists\text{gy}.\exists\text{gy}^-.\neg\text{Boldog}$, D tablója:

```
b •  {Boldog, ∃gy.∃gy⁻.¬Boldog}
  |
  |  gy
c •  {∃gy⁻.¬Boldog}
  |
  |  gy⁻
d •  {¬Boldog}
```

- Most építsünk tablót $D \sqcap \forall\text{gy}.\forall\text{gy}^-.\text{Szép}$-hez: a `c` pont címkéjében megjelenik $\forall\text{gy}^-.\text{Szép}$, a Szép fogalom lefelé és felfelé is terjed.

Inverz szerepek – szükséges változtatások

- A felfelé és lefelé ható szabályok miatt bevezetjük a *szomszéd* fogalmát
  - *y* az *x*-nek *R-szomszédja*, ha
    - *y* az *x*-nek *R*-követője vagy (1)
    - *x* az *y*-nak Inv(R)-követője. (2)
  - Példa (ism.):

$\mathtt{b}\ \bullet\ \{\text{Boldog}, \exists \text{gy}.\exists \text{gy}^-.\neg\text{Boldog}\}$
| gy
$\mathtt{c}\ \bullet\ \{\exists \text{gy}^-.\neg\text{Boldog}\}$
| gy^-
$\mathtt{d}\ \bullet\ \{\neg\text{Boldog}\}$

    - c-nek gy^--szomszédja: d, (1) miatt, valamint b, (2) miatt.
    - b-nek gy-szomszédja: c, (1) miatt
    - d-nek gy-szomszédja: c, (2) miatt
- A T-doboz lezárásában figyelembe kell venni az inverz szerepeket:
  - Ha **Trans**$(R) \in \mathcal{R}_\mathcal{T}$, akkor **Trans**(**Inv**$(R)) \in \mathcal{R}_\mathcal{T}$.
  - Ha $R \sqsubseteq S \in \mathcal{R}_\mathcal{T}$, akkor **Inv**$(R) \sqsubseteq$ **Inv**$(S) \in \mathcal{R}_\mathcal{T}$.

Inverz szerepek: egyenlőségi blokkolás

- A részhalmaz blokkolás inverz szerepek esetén nem megfelelő
- Kielégíthető-e a $(\forall \text{gy}^-.\neg\text{Okos}) \sqcap \text{Okos}$ fogalom $\{\top \sqsubseteq \exists \text{gy}.\text{Okos}\}$ felett?

$\mathtt{b}\ \bullet\ \{\forall \text{gy}^-.\neg\text{Okos}, \text{Okos}, \exists \text{gy}.\text{Okos}\}$
| gy
$\mathtt{c}\ \bullet\ \{\text{Okos}, \exists \text{gy}.\text{Okos}\}$

- c-t részhalmaz-blokkolja b, de ha c-t összevonjuk b-vel, rossz a modell:
  - Az átirányítás miatt b „kap” egy új gy^- kapcsolatot, aki Okos
  - Emiatt b már nem lesz példánya a $\forall \text{gy}^-.\neg\text{Okos}$ fogalomnak
- A megoldás egy erősebb blokkolásfajta, az ún. egyenlőségi blokkolás:
  - Az (egyenlőségi) blokkolás **alapfeltétele** fennáll egy *y* csúcs és annak egy *x* őse között ha $\mathcal{L}(y) = \mathcal{L}(x)$.
- Egyenlőségi blokkolás esetén a fenti c nem blokkolt, így létrejön egy d gy-követője, c-vel azonos címkével, amit c blokkol – jó modellt kapunk

Inverz szerepek: dinamikus blokkolás

- Még egy gond: a statikus blokkolás nem biztosítható, a blokkolt csúcs alatt is lehet részfa. Példa:
  kielégíthető-e *C* a $\{\top \sqsubseteq \exists R_1.\ldots.\exists R_n.\forall R_n^-.\ldots.\forall R_1^-.C\}$ T-doboz felett?
- Megoldás: a csomópontokat három osztályba soroljuk:
  - Közvetlenül blokkolt csúcsok: Egy *y* csomópont *közvetlenül blokkolt*, ha van olyan *x* őse, hogy *y*-ra és *x*-re fennáll a blokkolási alapfeltétel, de *y*-nak nincs két olyan őse, amelyek között ez a feltétel fennállna.
  - Közvetve blokkolt csúcsok: Egy *y* csomópont *közvetve blokkolt*, ha a van közvetlenül blokkolt őse.
  - Blokkolt csúcsok: Egy csomópont *blokkolt*, ha közvetlenül vagy közvetve blokkolt.
- A tabló-fa három szintje:
  - legfelül: nem blokkolt pontok: minden szabály alkalmazható
  - középen (határvonal): közvetlenül blokkolt csúcsok: a kiterjesztő szabályok nem alkalmazhatók
  - legalul: közvetve blokkolt csúcsok: semmilyen szabály sem alk.-ható

A $\mathcal{SHI}$ tabló transzformációs szabályai

$\sqcap$-szabály	
Feltétel:	$(C_1 \sqcap C_2) \in \mathcal{L}(x)$, x nem közvetve blokkolt, és $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$
T′ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.
$\sqcup$-szabály	
Feltétel:	$(C_1 \sqcup C_2) \in \mathcal{L}(x)$, x nem közvetve blokkolt, és $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$.
T$_1$ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_1\}$.
T$_2$ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_2\}$.
$\exists$-szabály	
Feltétel:	$(\exists R.C) \in \mathcal{L}(x)$, x nem blokkolt, és x-nek nincs olyan y R-szomszédja, amelyre $C \in \mathcal{L}(y)$.
T′ *új állapot:*	$V' = V \cup \{y\}$ (y egy új csomópont), $E' = E \cup \{\langle x, y \rangle\}$, $\mathcal{L}'(\langle x, y \rangle) = R$, $\mathcal{L}'(y) = \{C\}$.

A $\mathcal{SHI}$ tabló transzformációs szabályai (2)

$\forall$-szabály	
Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$, x nem közvetve blokkolt, és x-nek van egy olyan y R-szomszédja, amelyre $C \notin \mathcal{L}(y)$.
T′ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{C\}$.
$\forall_{+}$-szabály	
Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$, x nem közvetve blokkolt, van olyan S, amelyre $\mathrm{Trans}(S)$ és $S \sqsubseteq R \in \mathcal{R}_{\mathcal{T}}$, továbbá x-nek van egy olyan y S-szomszédja, amelyre $(\forall S.C) \notin \mathcal{L}(y)$.
T′ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{\forall S.C\}$.

A $\mathcal{SHI}$ nyelv – modellkonstrukció

- A közvetve blokkolt csomópontokat el kell hagyni:

$$\begin{aligned} \mathit{Base}_{\mathbf{T}}\ \mathcal{A} \;&=\; \{C(a) \mid C(a) \in \mathcal{A} \text{ és } a \text{ nem közvetve blokkolt } \mathbf{T}\text{-ben}\} \cup \\ & \quad\ \{R(a,b) \mid R(a,b) \in \mathcal{A} \text{ és } a, b \text{ nem közvetve blokkolt } \mathbf{T}\text{-ben}\} \end{aligned}$$

- Az inverz szerepekre nézve az adatdobozt le kell zárni:

$$\mathit{ClosI}\ \mathcal{A} \;=\; \mathcal{A} \cup \{\mathrm{Inv}(R)(b,a) \mid R(a,b) \in \mathcal{A}\}$$

- A **T** tablóhoz tartozó modell:

$$\mathcal{I}_{\mathbf{T}} = \mathcal{I}^{nat}(\mathit{ClosHT}_{\mathcal{T}}\ \mathit{ClosI}\ \mathit{IdBlock}_{\mathbf{T}}\ \mathit{Base}_{\mathbf{T}}\ A_{\mathbf{T}})$$

A funkcionális korlátozások

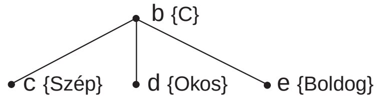
- Az $\mathcal{F}$ nyelvkiterjesztés: a $(\leqslant 1\ R)$ és $(\geqslant 2\ R)$ fogalomkifejezések ($\mathcal{N}$-nél gyengébb)
- Példa: vizsgáljuk a $\forall\text{gyereke}^{-}.\bot$ (nincs szülője) fogalom kielégíthetőségét a $\{\top \sqsubseteq \exists\text{gyereke}.\top \sqcap (\leqslant 1\ \text{gyereke}^{-})\}$ (mindenkinek van gyereke, és legfeljebb 1 szülője van)T-doboz felett.
  - Ezt a feladat kielégíthető, de csak végtelen modellel
  - Ha volna egy véges, pl. n-elemű modell, akkor ebben
    - legalább n gyereke-kapcsolat lenne (mert mindenkinek van gyereke);
    - de ugyanakkor legfeljebb $n-1$ darab gyereke^{-}-kapcsolat lehetne (mert mindenkinek legfeljebb egy szülője van, és a „nincs szülője” fogalom kielégíthető, tehát van legalább egy olyan egyed akinek 0 szülője van).
    - Ez ellentmondás.

Végtelen modellek építése bemásolással

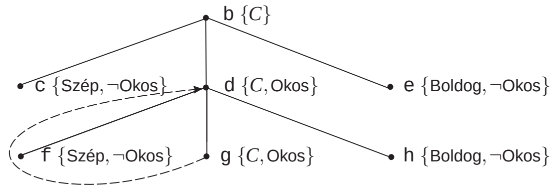
- Példa: vizsgáljuk C kielégíthetőségét, az alábbi T-doboz felett:

$$\begin{aligned} C \;&\equiv\; \exists\text{gy}.\text{Szép} \sqcap \exists\text{gy}.\text{Okos} \sqcap \exists\text{gy}.\text{Boldog} \sqcap (\leqslant 1\ \text{gy}^{-}) && (1) \\ \text{Okos} \;&\sqsubseteq\; C && (2) \end{aligned}$$

  - C önmagában vett tablója:



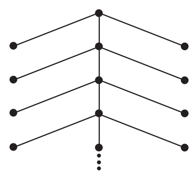
  - A teljes példa-tabló; (2) belsősítését $(\neg\text{Okos} \sqcup C)$, hozzávesszük minden csúcshoz, majd alkalmazzuk az $\sqcup$-szabályt:

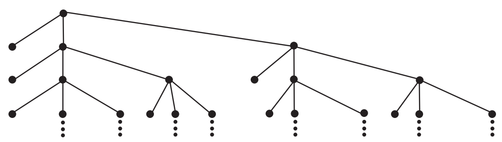


  - A g csúcsot blokkolja a d csúcs.

Végtelen modellek építése bemásolással (2)

- Bemásolás:
  - A blokkoló csúcs alatti teljes fát bemásoljuk a blokkolt csúcsba.
  - A blokkolt csúcsok másolataira ismételjük ezt a folyamatot, a végtelenségig.
- A bemásolásos modell a korábbi példára (bal oldal) illetve azt a Boldog $\sqsubseteq C$ axiómával kiegészítve (jobb oldal):





Blokkolás bemásolásos modellkonstrukció esetén

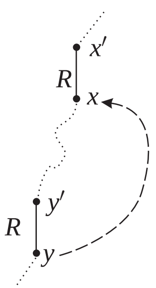
- Példa ($\mathcal{SHI}$): Az Okos fogalom kielégíthető-e a $\{\top \sqsubseteq C_0\}$ T-doboz felett, ahol $C_0 = \exists$gy.$\exists$gy$^-$.Okos (van okos szülővel rendelkező gyereke)?
- Egyenlőségi blokkolást alkalmazva `d`-t blokkolja a `c` csúcs:

$$
\begin{array}{ll}
\mathtt{b} \bullet & \{C_0,\ \text{Okos}\} \\
\quad | & \text{gy} \\
\mathtt{c} \bullet & \{C_0,\ \exists\text{gy}^-.\text{Okos}\} \\
\quad | & \text{gy} \\
\mathtt{d} \bullet & \{C_0,\ \exists\text{gy}^-.\text{Okos}\}
\end{array}
$$

- Átirányítással: ($\mathtt{d} \mapsto \mathtt{c}$): $\Delta^{\mathcal{I}} = \{\mathtt{b}, \mathtt{c}\}$, Okos$^{\mathcal{I}} = \{\mathtt{b}\}$, gy$^{\mathcal{I}} = \{\langle \mathtt{b}, \mathtt{c}\rangle, \langle \mathtt{c}, \mathtt{c}\rangle\}$
- Bemásolással:`b` gy $\mathtt{c}_1$ gy $\mathtt{c}_2$ gy $\ldots$ gy $\mathtt{c}_n$ gy $\mathtt{c}_{n+1}$ gy $\ldots$
  - csak `b` Okos, $\{\top \sqsubseteq C_0\}$ nyilván nem teljesül.
  - A blokkolt csúcs megörökli az alsó szomszédokat, de a felsőt nem!
- A példa $\mathcal{SHIF}$ változata: Okos($\sqcap\forall$gy$^-.\bot$) kielégíthető-e $\{\top \sqsubseteq C_1\}$ felett, ahol $C_1 = C_0 \sqcap (\leqslant 1\,\text{gy}^-)$ (legfeljebb 1 szülője van)
  - Itt az átirányítás nem jó, csak a bemásolás
  - A megoldás: az ún. páros blokkolás biztosítja, hogy a felső szomszéd is megfelelő legyen

A páros blokkolás definíciója

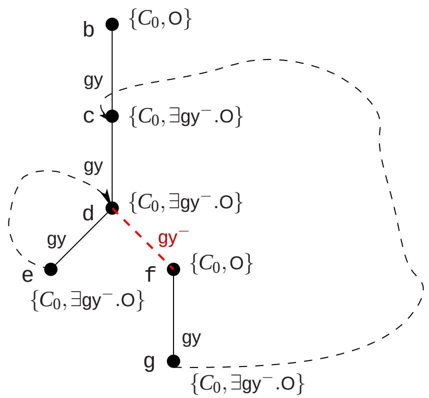
- **A blokkolási alapfeltétel** (*páros blokkolásra*) Azt mondjuk, hogy egy y csomópont és annak egy x őse között fennáll a blokkolási alapfeltétel, ha x-nek van egy x' megelőzője és y-nak a megelőzője y', továbbá ezen csomópontokra fennállnak a következő állítások:
  - $\mathcal{L}(y) = \mathcal{L}(x)$
  - $\mathcal{L}(y') = \mathcal{L}(x')$
  - $\mathcal{L}(\langle y', y\rangle) = \mathcal{L}(\langle x', x\rangle)$

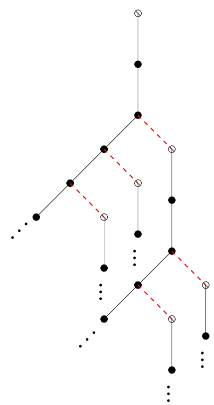


- A közvetlen, ill. közvetett blokkolás definíciója változatlan formában épül a blokkolási alapfeltételre

Példa páros blokkolásra

- Alkalmazzunk *páros* blokkolást a korábbi példában! Az Okos fogalom kielégíthető-e a $\{\top \sqsubseteq C_0\}$ T-doboz felett, ahol $C_0 = \exists$gy.$\exists$gy$^-$.Okos?
- Bal oldal: a teljes tabló (`e`-t blokkolja `d` és `g`-t blokkolja `c`)
- Jobb oldal: a bemásolásos model (a folytonos élek címkéje gy, a szaggatottaké gy$^-$, az utóbbiak alsó végpontja és a fa gyökere tartozik az Okos fogalomba)





További változások a $\mathcal{SHIF}$ tabló-algoritmusban

- Összevonás kell ha $(\leqslant 1\, R) \in \mathcal{L}(x)$ és x-nek 2 R-szomszédja van, y és z.
- Az y-ba és z-be vezető élek címkéi különbözhetnek (szerephierarchia):
  - x-nek rokona-követője y és x-nek barátja-követője z
  - $(\leqslant 1\,\text{ismerőse}) \in \mathcal{L}(x)$, barátja $\sqsubseteq$ ismerőse és rokona $\sqsubseteq$ ismerőse.
- Megoldás: az éleket (esetleg inverz) szerepek *halmazával* címkézzük
- Emiatt (triviálisan) változik a „követő” fogalma:
  - y az x *R-követője*, ha $\exists S \sqsubseteq R \in \mathcal{R}_\mathcal{T}$, $y \in V$, hogy $S \underline{\in} \mathcal{L}(\langle\, x, y\,\rangle)$.
- Új transzformációs szabályok:
  - $\leqslant$-szabály
    - két eset: egy felső és egy alsó szomszéd (az alsó szűnik meg), vagy két alsó szomszéd (az egyik megszűnik)
    - a megmaradó csomópont/él megkapja a megszüntetett címkéit
    - a megszüntetett él címkéje $\emptyset$ lesz, ez közvetett blokkolást jelent
    - most nem foglalkozunk a megszüntetett csúcsból induló élek „áthozatalával” – szükség esetén majd újra létrejönnek (szemben az $\mathcal{ALCN}$ tabló-algoritmussal)
  - $\geqslant$-szabály: két követőt hoz létre, az egyik A, a másik $\neg A$ címkét kap (A „új”, másutt elő nem forduló atomi fogalom)

A $\mathcal{SHIF}$ tabló-algoritmus $\exists$-szabálya

- Csak annyi a változás, hogy az élet szerephalmazzal címkézzük:

$\exists$-szabály

Feltétel:	$(\exists R.C) \in \mathcal{L}(x)$, x nem blokkolt, és
	x-nek nincs olyan y R-szomszédja, amelyre $C \in \mathcal{L}(y)$.
T′ *új állapot:*	$V' = V \cup \{y\}$ (y egy új csomópont),
	$E' = E \cup \{\langle\, x, y\,\rangle\}$, $\mathcal{L}'(\langle\, x, y\,\rangle) = \underline{\{}R\underline{\}}$, $\mathcal{L}'(y) = \{C\}$.

Emlékeztető: színezéssel és aláhuzással jelezzük a módosult részeket.

A $\mathcal{SHIF}$ tabló-algoritmus új szabályai

$\geqslant$-szabály

Feltétel:	$(\geqslant 2\, R) \in \mathcal{L}(x)$, x nem blokkolt, és
	x-nek nincs olyan y R-követője, amelyre $A \in \mathcal{L}(y)$.
T′ *új állapot:*	$V' = V \cup \{y, z\}$ (y_i új csomópontok),
	$E' = E \cup \{\langle\, x, y\,\rangle, \langle\, x, z\,\rangle\}$,
	$\mathcal{L}'(\langle\, x, y\,\rangle) = \{R\}, \mathcal{L}'(\langle\, x, z\,\rangle) = \{R\}$,
	$\mathcal{L}'(y) = \{A\}, \mathcal{L}'(z) = \{\neg A\}$

- A fenti szabályban A egy olyan atomi fogalom, amely a vizsgált tudásbázisban nem fordul elő
- Itt még nem használjuk a tabló-állapot I komponensét (egyenlőtlenségek halmaza). Az össze-nem-vonhatóságot úgy érjük el, hogy a létrehozott két követő egyikének címkéjébe A, a másikába $\neg A$ kerül

A $\mathcal{SHIF}$ tabló-algoritmus új szabályai (2)

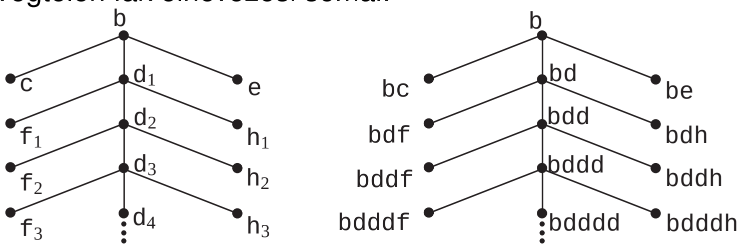
$\leqslant$-szabály

Feltétel:	$(\leqslant 1\, R) \in \mathcal{L}(x)$, x nem közvetetten blokkolt, és	
	x-nek van két R-szomszédja, y és z,	
	ahol $y \not\Rightarrow x$	(1)
T′ *új állapot:*	$\mathcal{L}'(z) = \mathcal{L}(z) \cup \mathcal{L}(y)$	
	$\mathcal{L}'(\langle\, x, y\,\rangle) = \emptyset$,	
	$\mathcal{L}'(\langle\, z, x\,\rangle) = \mathcal{L}(\langle\, z, x\,\rangle) \cup \mathrm{Inv}^*(\mathcal{L}(\langle\, x, y\,\rangle))$, ha $z \Rightarrow x$	(2)
	$\mathcal{L}'(\langle\, x, z\,\rangle) = \mathcal{L}(\langle\, x, z\,\rangle) \cup \mathcal{L}(\langle\, x, y\,\rangle)$, ha $z \not\Rightarrow x$	(3)

- Jelölések:
  - $a \Rightarrow b$: a-nak követője b; $a \not\Rightarrow b$: a-nak nem követője b
  - Inv^* halmazfüggvény: az Inv függvény elemenkénti alkalmazása:
$$\mathrm{Inv}^*(H) = \{\mathrm{Inv}(R) | R \in H\}$$
- Az (1) „rejtvény” megfejtése: y nem felső szomszéd, azaz: ha a két szomszéd között van felső, akkor azt jelöljük z-vel
- A (2) feltétel értelmezése: van (z-vel jelölt) felső szomszéd
- A (3) feltétel értelmezése: y és z mindketten alsó szomszédok

A $\mathcal{SHIF}$ tabló modellkonstrukciója: a bemásolás formális leírása

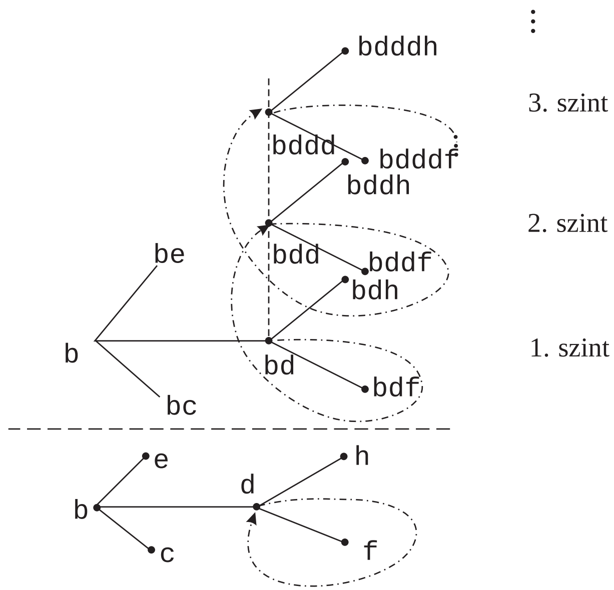
- A végtelen fák elnevezési sémái:



  - A példában számozhatjuk az ismétlődő csúcsokat (bal oldal)
  - Általánosan a végtelen fa csúcsait az átirányításos tabló-gráfban a gyökérpontból kiinduló utakkal címkézzük (jobb oldal):
- Az átirányításos tabló-gráf éle $\langle x, z \rangle$ csakkor ha:
  - van egy $\langle x, z \rangle \in E$ él, ahol z nem blokkolt.
    A példában ezek: $\langle \mathtt{b}, \mathtt{c} \rangle, \langle \mathtt{b}, \mathtt{d} \rangle, \langle \mathtt{b}, \mathtt{e} \rangle, \langle \mathtt{d}, \mathtt{f} \rangle, \langle \mathtt{d}, \mathtt{h} \rangle$.
  - van egy $\langle x, y \rangle \in E$ él, ahol y-t blokkolja z.
    A példában $\langle \mathtt{d}, \mathtt{g} \rangle \in E$, $\mathtt{g}$-t blokkolja $\mathtt{d}$, tehát a $\langle \mathtt{d}, \mathtt{d} \rangle$ hurokél ilyen
- Az átirányításos tabló-gráf éle $\langle x, z \rangle$:
  ezt formálisan „*x örököse z*” kapcsolatként definiáljuk hamarosan.

A $\mathcal{SHIF}$ tabló modellkonstrukciója (2)

- A végtelen fastruktúra spirálos szemléltetése



A $\mathcal{SHIF}$ tabló modellkonstrukciója (3)

- A végtelen fastruktúra formális építése
  - egy **T** tablóban egy $x \in V$ csomópontnak R-örököse egy $y \in V$ csomópont, ha sem x sem y nem blokkolt, és
    - vagy $R \in \mathcal{L}(\langle x, y \rangle)$, azaz x-ből y-ba egy R címkéjű él vezet;
    - vagy van olyan z, hogy y blokkolja z-t és $R \in \mathcal{L}(\langle x, z \rangle)$, azaz x-ből egy y által blokkolt csúcsba vezet egy R címkéjű él.
  - x-nek örököse y, ha van olyan R, hogy x-nek R-örököse y.
- A végtelen adatdoboz egyedeinek halmaza:

$$\begin{aligned} Dom^{cp}_{\mathbf{T}} \;=\; \{[x_0, \ldots, x_n] \mid & \;\mathbf{T} \text{ gyökérpontja } x_0,\ x_0\text{-nak örököse } x_1, \\ & x_1\text{-nek örököse } x_2,\ \ldots,\ x_{n-1}\text{-nek örököse } x_n\} \end{aligned}$$

- Jelölések: ha $X = [x_0, ..., x_n]$, $last(X) = x_n$, $X \oplus Y$ két lista összefűzése
- A végtelen adatdoboz:

$$\begin{aligned} \mathcal{A}^{cp}_{\mathbf{T}} \;=\; & \{C(X) \mid X \in Dom^{cp}_{\mathbf{T}}, last(X) = x, \text{ és } C \in \mathcal{L}(x)\} \cup \\ & \{R(X, Y) \mid X, Y \in Dom^{cp}_{\mathbf{T}}, X \oplus [y] = Y, \text{ és } last(X)\text{-nek } R\text{-örököse } y\} \end{aligned}$$

- A lezárásokkal $\mathcal{A}^{cp}_{\mathbf{T}}$-ből önmegvalósító adatdoboz lesz, így a modell:

$$\mathcal{I}_{\mathbf{T}} = \mathcal{I}^{nat}(ClosHT_{\mathcal{T}}\ ClosI\ A^{cp}_{\mathbf{T}})$$

Minősített számosság-korlátozások – a $\mathcal{Q}$ nyelvkiterjesztés

- Hasonló az $\mathcal{N}$ kiterjesztéshez, adott C fogalomhoz tartozó R-követők számát korlátozhatjuk: $(\leqslant n\, R.C)$, $(\geqslant n\, R.C)$
- A $\leqslant$- és $\geqslant$-szabályok is csak ennyiben változnak:
  - A $\leqslant$-szabály egy $(\leqslant n\, R.C)$ fogalomra akkor tüzel, ha talál $n + 1$ olyan R-szomszédot, amelyek mindegyike rendelkezik a C címkével, és ekkor elvégez egy megfelelő összevonást.
  - A $\geqslant$-szabály egy $(\geqslant n\, R.C)$ fogalomra akkor tüzel, ha *nem* talál n olyan R-szomszédot, amelyek mind rendelkeznek a C címkével, és ekkor n darab C címkéjű nem-összevonható R-követőt hoz létre.
- Új ütközési feltétel: ha x címkéjében szerepel $(\leqslant n\, R.C)$, és x-nek van $n + 1$ darab C címkéjű szomszédja, amelyek között nincs két összevonható.

A választási szabály

- Példa: az alábbi fogalom nyilván nem kielégíthető:

$$(\geqslant 3\mathrm{gy}.\top) \sqcap (\leqslant 1\,\mathrm{gy}.\mathrm{Szép}) \sqcap (\leqslant 1\,\mathrm{gy}.\neg\mathrm{Szép})$$

  Az eddig ismertetett szabályok ezt nem mutatják ki, szükség van az alábbi új szabályra is.
- $\bowtie$-szabály (*választási* szabály): ha egy x csúcs címkéjében szerepel a $(\bowtie nR.C)$ fogalom (ahol $\bowtie$ a $\leqslant$ és $\geqslant$ egyikét jelöli)
  - akkor a csúcs minden követőjéhez nemdeterminisztikus módon hozzávesszük egyrészt a C címkét, másrészt ennek negáltját
  - pontosabban: $\neg C$ negációs normálalakját, amit $\sim C$-vel jelölünk

Tartalom

A $\mathcal{SHIQ}$ tablóhoz szükséges negációs normálalak

- Egy tetszőleges $\mathcal{SHIQ}$ fogalom negációs normálalakja előállítható az alábbi átalakítási szabályok ismételt alkalmazásával, amíg csak ez lehetséges:

$$\begin{array}{rcl}
\neg\neg C & \Rightarrow & C \\
\neg(C \sqcap D) & \Rightarrow & \neg C \sqcup \neg D \\
\neg(C \sqcup D) & \Rightarrow & \neg C \sqcap \neg D \\
\neg(\exists R.C) & \Rightarrow & \forall R.\neg C \\
\neg(\forall R.C) & \Rightarrow & \exists R.\neg C \\
\neg(\leqslant n\,R.C) & \Rightarrow & (\geqslant m\,R.C) \text{ ahol } m = n+1 \\
\neg(\geqslant 1\,R.C) & \Rightarrow & \forall R.\neg C \\
\neg(\geqslant n\,R.C) & \Rightarrow & (\leqslant m\,R.C) \text{ ahol } m = n-1 \text{ és } n > 1
\end{array}$$

A $\mathcal{SHIQ}$ tabló: előzetes definíciók

- Legyen C egy negációs normálalakra hozott tetszőleges $\mathcal{SHIQ}$ fogalomkifejezés, $\mathcal{T}$ egy csak szerepaxiómákat tartalmazó T-doboz.
- A feladat annak eldöntése, hogy C kielégíthető-e $\mathcal{T}$ felett.
- A $\mathcal{T}$ T-dobozból képezzük annak $\mathcal{R}_\mathcal{T}$ tranzitív-reflexív lezárását, mint azt a legszűkebb halmazt, amelyre:
  - Minden $R \equiv S \in \mathcal{T}$ esetén $R \sqsubseteq S \in \mathcal{R}_\mathcal{T}$ és $S \sqsubseteq R \in \mathcal{R}_\mathcal{T}$ egyaránt fennáll.
  - Minden $T \in \mathcal{T}$ esetén $T \in \mathcal{R}_\mathcal{T}$, ahol T vagy $R \sqsubseteq S$, vagy **Trans**(**R**) alakú.
  - Ha R egy $\mathcal{T}$-ben előforduló szerepnév, vagy $R \in$ *roles*(C), akkor $R \sqsubseteq R \in \mathcal{R}_\mathcal{T}$
  - Ha **Trans**$(R) \in \mathcal{R}_\mathcal{T}$, akkor **Trans**(**Inv**$(R)) \in \mathcal{R}_\mathcal{T}$.
  - Ha $R \sqsubseteq S \in \mathcal{R}_\mathcal{T}$, akkor **Inv**$(R) \sqsubseteq$ **Inv**$(S) \in \mathcal{R}_\mathcal{T}$.
  - Ha $R \sqsubseteq R' \in \mathcal{R}_\mathcal{T}$, és $R' \sqsubseteq R'' \in \mathcal{R}_\mathcal{T}$, akkor $R \sqsubseteq R'' \in \mathcal{R}_\mathcal{T}$.

A tabló-állapot

- Egy D fogalom esetén $subneg(D)$ jelölje azt a legszűkebb halmazt, amely tartalmazza D-t és zárt a részkifejezés-képzésre és a $\sim$ műveletre nézve, ahol $\sim C = \neg C$ NNF alakja
  - A tabló-gráf csúcsait $subneg(C)$ részhalmazaival címkézzük.
- Definíció: $roles(D)$ a D fogalomban előforduló (esetleg inverz) szerepkifejezések halmaza
  - A tabló-gráf éleit $roles(C)$ részhalmazaival címkézzük.
- **T** tabló-állapot: egy $\langle V, E, \mathcal{L}, I \rangle$ négyes, ahol
  - a $\langle V, E, \mathcal{L} \rangle$ hármas egy címkézett irányított gráf:
    - V a csomópontok halmaza
    - $E \subseteq V \times V$ az élek halmaz
    - $\mathcal{L}$ a csomópontokhoz és élekhez *címkéket* rendelő függvény: $\mathcal{L}(x) \subseteq subneg(C)$, $\mathcal{L}(\langle x, y \rangle) \subseteq roles(C)$
  - I egy olyan halmaz, amely $x \neq y$ alakú egyenlőtlenségekből áll, ahol $x, y \in V$ a gráf csúcsai.

$\mathcal{SHIQ}$ tabló: a gráf szerkezete

- Egy y csomópont az x *R-követője*, ha van olyan $S \in \mathcal{L}(\langle x, y \rangle)$ amelyre $S \sqsubseteq R \in \mathcal{R}_{\mathcal{T}}$. Ebben az esetben azt is mondjuk, hogy y az x követője.
- Egy x csomópont az y csúcs *megelőzője*, ha y az x-nek követője.
- Egy x csomópont a z csúcs *őse*, ha x a z-nek megelőzője, vagy ha z-nek van olyan y őse, amelynek megelőzője x.
- Egy z csomópont az x csúcsnak *leszármazottja*, ha x az z-nek őse.
- y az x-nek *R-szomszédja*, ha x-nek R-követője, vagy x az y-nak $\mathrm{Inv}(R)$-követője.

$\mathcal{SHIQ}$ tabló: blokkolás

- Azt mondjuk, hogy egy y csomópont és annak egy x őse között fennáll a *(páros) blokkolási alapfeltétel*, ha x-nek van egy x' megelőzője és y-nak a megelőzője y', továbbá ezen csomópontokra fennállnak a következő állítások:
  - $\mathcal{L}(y) = \mathcal{L}(x)$
  - $\mathcal{L}(y') = \mathcal{L}(x')$
  - $\mathcal{L}(\langle y', y \rangle) = \mathcal{L}(\langle x', x \rangle)$
- Egy y csomópont *közvetlenül blokkolt*, ha van olyan x őse, hogy y-ra és x-re fennáll a blokkolási alapfeltétel, de y-nak nincs két olyan őse, amelyekre ez a feltétel fennállna. Ebben az esetben azt is mondjuk, hogy x *közvetlenül blokkolja* y-t.
- Egy y csomópont *közvetve blokkolt*, ha van blokkolt őse, vagy ha y az x csúcs követője és $\mathcal{L}(\langle x, y \rangle) = \emptyset$ (ez utóbbi eset azt jelzi, hogy az y csúcs megszűnt, mivel az algoritmus során egy másik csomóponttal összevontuk).
- Egy csomópont *blokkolt*, ha közvetlenül vagy közvetve blokkolt.

$\mathcal{SHIQ}$ tabló: Ütközések

- Az x és y csomópontokat *összevonhatónak* mondjuk, ha $x \neq y \notin I$ és $y \neq x \notin I$.
- Egy tabló-állapot *ütközést* tartalmaz, ha a tabló-gráfnak van egy olyan x pontja, amelyre
  - $\bot \in \mathcal{L}(x)$, vagy
  - $\{C, \neg C\} \subseteq \mathcal{L}(x)$, vagy
  - $(\leqslant n\, R.C) \in \mathcal{L}(x)$ és x-nek van $n+1$ darab olyan $y_0, \ldots, y_n$ R-szomszédja, amelyek mindegyikére $C \in \mathcal{L}(y_i)$, $i = 0, \ldots n$, és amelyek között nincs két összevonható.

A $\mathcal{SHIQ}$ tabló-algoritmus és transzformációs szabályai

- A kiindulási tabló-állapot: $\mathbf{T}_0 = \langle \{x_0\}, \emptyset, (\mathcal{L}(x_0) \mapsto \{C\}), \emptyset \rangle$, ahol x_0 a gyökérpont.
- Egy tabló-állapot teljes, ha semelyik szabály nem alkalmazható rá.
- Ha az alábbi szabályok ismételt alkalmazásával egy teljes és ütközésmentes tablóhoz jutunk, akkor az algoritmus válasza: „C kielégíthető $\mathcal{T}$ felett”.
- Ha minden tabló-állapot ütközést tartalmaz, a válasz: „C nem kielégíthető $\mathcal{T}$ felett”.

A metszet- és unió-szabályok

$\sqcap$-szabály	
Feltétel:	$(C_1 \sqcap C_2) \in \mathcal{L}(x)$, x nem közvetve blokkolt, és $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$
$\mathbf{T}'$ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_1, C_2\}$.
$\sqcup$-szabály	
Feltétel:	$(C_1 \sqcup C_2) \in \mathcal{L}(x)$, x nem közvetve blokkolt, és $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$.
$\mathbf{T}_1$ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_1\}$.
$\mathbf{T}_2$ *új állapot:*	$\mathcal{L}'(x) = \mathcal{L}(x) \cup \{C_2\}$.

A létezik- és minden-szabályok

$\exists$-szabály	
Feltétel:	$(\exists R.C) \in \mathcal{L}(x)$, x nem blokkolt, és x-nek nincs olyan y R-szomszédja, amelyre $C \in \mathcal{L}(y)$.
$\mathbf{T}'$ *új állapot:*	$V' = V \cup \{y\}$ (y egy új csomópont), $E' = E \cup \{\langle x, y \rangle\}$, $\mathcal{L}'(\langle x, y \rangle) = \{R\}$, $\mathcal{L}'(y) = \{C\}$.
$\forall$-szabály	
Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$, x nem közvetve blokkolt, és x-nek van egy olyan y R-szomszédja, amelyre $C \notin \mathcal{L}(y)$.
$\mathbf{T}'$ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{C\}$.
$\forall_+$-szabály	
Feltétel:	$(\forall R.C) \in \mathcal{L}(x)$, x nem közvetve blokkolt, van olyan S, amelyre $\mathrm{Trans}(S) \in \mathcal{R}_\mathcal{T}$, $S \sqsubseteq R \in \mathcal{R}_\mathcal{T}$, és x-nek van y S-szomszédja, amelyre $(\forall S.C) \notin \mathcal{L}(y)$.
$\mathbf{T}'$ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{\forall S.C\}$.

A számosság-korlátozásokra vonatkozó szabályok

$\bowtie$- szabály	
Feltétel:	$(\bowtie nR.C) \in \mathcal{L}(x)$, x nem közvetve blokkolt, és x-nek van olyan y R-szomszédja, hogy $\{C, \sim C\} \cap \mathcal{L}(y) = \emptyset$
$\mathbf{T}_1$ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{C\}$.
$\mathbf{T}_2$ *új állapot:*	$\mathcal{L}'(y) = \mathcal{L}(y) \cup \{\sim C\}$.
$\geqslant$-szabály	
Feltétel:	$(\geqslant n\, R.C) \in \mathcal{L}(x)$, x nem blokkolt, és nincs n darab olyan $y_1, \ldots, y_n$ csúcs, amelyek páronként nem összevonhatóak, valamint minden i-re x-nek y_i R-szomszédja és $C \in \mathcal{L}(y_i)$.
$\mathbf{T}'$ *új állapot:*	$V' = V \cup \{y_1, \ldots, y_n\}$ (y_i új csomópontok), $E' = E \cup \{\langle x, y_1 \rangle, \ldots, \langle x, y_n \rangle\}$, $\mathcal{L}'(\langle x, y_i \rangle) = \{R\}$, $\mathcal{L}'(y_i) = \{C\}$, minden $i = 1 \leq i \leq n$, $I' = I \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n\}$.

A számosság-korlátozásokra vonatkozó szabályok (2)

$\leqslant$-szabály

Feltétel: $(\leqslant n\,R.C) \in \mathcal{L}(x)$, és x nem közvetlenül blokkolt, és x-nek van $n+1$ R-szomszédja, $y_0, \ldots, y_n$, amelyekre $C \in \mathcal{L}(y_i)$, és amelyek között van két összevonható.

Minden $(0 \leq i < j \leq n)$-re ahol y_i és y_j összevonható, legyen $\{y, z\} = \{y_i, y_j\}$ úgy, hogy y-nak x nem követője:

$\mathbf{T}_{ij}$ *új állapot:* $\mathcal{L}'(z) = \mathcal{L}(z) \cup \mathcal{L}(y)$

$\mathcal{L}'(\langle x, y \rangle) = \emptyset$,

$\mathcal{L}'(\langle z, x \rangle) = \mathcal{L}(\langle z, x \rangle) \cup \text{Inv}^*(\mathcal{L}(\langle x, y \rangle))$, ha $z \Rightarrow x$

$\mathcal{L}'(\langle x, z \rangle) = \mathcal{L}(\langle x, z \rangle) \cup \mathcal{L}(\langle x, y \rangle)$, ha $z \not\Rightarrow x$

$I' = I[y \to z]$ (y minden előfordulását z-re cseréljük).

Jelölések

$z \Rightarrow x$: z-nek követője x

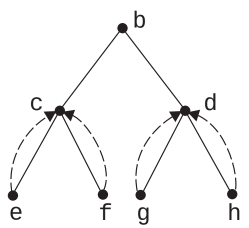
$$\text{Inv}^*(H) = \{\text{Inv}(R) | R \in H\}$$

A $\mathcal{SHIQ}$ tabló-algoritmus tulajdonságai

- Terminálás
  - Jelöljük: $m = |subneg(C)|$, $n = |roles(C)|$.
  - A gyökérpontból induló $2^{2m+n} + 2$ hosszú úton van olyan y pont, amelyet egy x őse blokkol,
    - az összes egymást követő u és v pontpár esetén a különböző $\langle \mathcal{L}(\langle u, v \rangle), \mathcal{L}(u), \mathcal{L}(v) \rangle$ hármasok száma maximum 2^{2m+n} lehet.
  - A tabló-fa elágazási szélessége korlátos (mint $\mathcal{ALCN}$-ben)
  - A tabló-fa adatdoboza bővül (mint $\mathcal{ALCN}$-ben) – nem lehet ciklus
- Teljesség: ha C kielégíthető, akkor az algoritmus ezt jelzi (vö. $\mathcal{ALCN}$):
  - Átfogalmazás: ha a tabló-algoritmus nem-kielégíthetőséget jelez, akkor a fogalom nem kielégíthető.
  - Minden $\mathbf{T}$ tablónak megfelel egy $\mathcal{A}_{\mathbf{T}}$ adatdoboz.
  - Az egyes transzformációs szabályok egy $\mathbf{T}$ tablót egy $S_{\mathbf{T}}$ tablóhalmazba visznek át, úgy hogy $\mathcal{A}_{\mathbf{T}}$ akkor és csak akkor kielégíthető, ha van olyan $\mathbf{T}' \in S_{\mathbf{T}}$, hogy $\mathcal{A}_{\mathbf{T}'}$ kielégíthető.
- Helyesség: ha a tabló-algoritmus kielégíthetőséget jelez, akkor a fogalom kielégíthető – modellkonstrukció

A $\mathcal{SHIQ}$ tabló modellkonstrukciója – példa

- Vizsgáljuk a $\top$ fogalom kielégíthetőségét a $\mathcal{T}_0 = \{\top \sqsubseteq C_0\}$ T-doboz felett, ahol $C_0 = (\geqslant 2\,\text{gy}) \sqcap (\leqslant 1\,\text{gy}^-)$.
- A teljes és ütközésmentes tabló-állapot – minden él címkéje $\{\text{gy}\}$, minden csúcs címkéje $\{C_0, (\geqslant 2\,\text{gy}), (\leqslant 1\text{gy}^-)\}$:



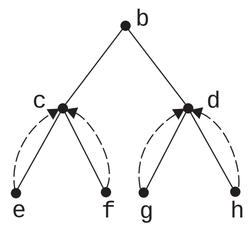
- a $\mathcal{SHIF}$ tablónál használt módszerrel az alaphalmaz a

$$\{[b], [b, c], [b, d], [b, c, c], [b, d, d], \ldots\}$$

  alakú listákból állna.
  - Ez nem jó, mert nincs minden csúcsnak két gy-követője.

A $\mathcal{SHIQ}$ tabló modellkonstrukciója – példa

- A teljes és ütközésmentes tabló-állapot (ismétlés):



- A megoldás: vegyük bele a blokkolt csúcsot is az útvonalba, pl. a $[b, c]$ egyed két gy-követője $[b, c, e, c]$ és $[b, c, f, c]$ legyen
- Technikai változtatás: párok listája (a nem blokkoltak duplázva): pl. a $[\langle b, b \rangle, \langle c, c \rangle]$ két gy-követője: $[\langle b, b \rangle, \langle c, c \rangle, \langle c, e \rangle], [\langle b, b \rangle, \langle c, c \rangle, \langle c, f \rangle]$

A $\mathcal{SHIQ}$ tabló modellkonstrukciója

- A végtelen fastruktúra formális építése
  - egy **T** tablóban egy $x \in V$ csomópontnak R-örököse egy $\langle y, z \rangle$ pontpár, ahol $y, z \in V$, ha sem x sem y nem blokkolt, és
    - vagy $R \in \mathcal{L}(\langle x, y \rangle)$, azaz x-ből y-ba egy R címkéjű él vezet, és $z = y$;
    - vagy y blokkolja z-t és $R \in \mathcal{L}(\langle x, z \rangle)$, azaz x-ből az y csomópont által blokkolt z csúcsba vezet egy R címkéjű él.
  - x-nek örököse $\langle y, z \rangle$, ha van olyan R, hogy x-nek R-örököse $\langle y, z \rangle$.
- A végtelen adatdoboz egyedeinek halmaza:

$$Dom^{cpQ}_{\mathbf{T}} = \{[\langle x_0, x_0 \rangle, \langle x_1, u_1 \rangle, \ldots, \langle x_n, u_n \rangle] \mid \mathbf{T} \text{ gyökere } x_0, x_0 \text{ örököse } \langle x_1, u_1 \rangle, x_1 \text{ örököse } \langle x_2, u_2 \rangle, \ldots, x_{n-1} \text{ örököse } \langle x_n, u_n \rangle\}$$

A $\mathcal{SHIQ}$ tabló modellkonstrukciója (2)

- Jelölések (legyen $X = [\langle x_0, u_0 \rangle, \ldots, \langle x_n, u_n \rangle]$):
  - $last_1(X)$ az X lista utolsó elemének első tagja, azaz x_n,
  - $X \oplus Y$ az X és Y lista egymás után fűzése
- Az $\mathcal{A}^{cpQ}_{\mathbf{T}}$ végtelen adatdoboz állításhalmaza:
  $\{C(X) \mid X \in Dom^{cpQ}_{\mathbf{T}}, last_1(X) = x, \text{ és } C \in \mathcal{L}(x)\}$
  $\cup\{R(X, Y) \mid X, Y \in Dom^{cpQ}_{\mathbf{T}}, X \oplus [\langle y, u \rangle] = Y, \ last_1(X)\ R\text{-örököse } \langle y, u \rangle\}$
- Az inverz szerepekre, szerephierarchiára és tranzitivitásra vonatkozó lezárás után önmegvalósító adatdobozt kapunk
- Tehát C-nek $\mathcal{T}$ feletti modellje az alábbi természetes interpretáció:

$$\mathcal{I}_{\mathbf{T}} = \mathcal{I}^{nat}(ClosHT_{\mathcal{T}}\ ClosI\ A^{cpQ}_{\mathbf{T}})$$

Az adatdobozos $\mathcal{SHIQ}$ tabló-algoritmus

- Tegyük fel, hogy az $\mathcal{A}$ adatdoboz egyedneveinek halmaza $\{a_1, \ldots, a_n\}$.
- A $\mathbf{T}_0 = \langle V_0, E_0, \mathcal{L}_0, I_0 \rangle$ kezdeti tabló-állapot:

$$\begin{aligned} V_0 &= \{a_1, \ldots, a_n\} \\ E_0 &= \{\langle a, b \rangle \mid \text{ van olyan } R, \text{ hogy } R(a, b) \in \mathcal{A}\} \\ \mathcal{L}_0(a) &= \{C \mid C(a) \in \mathcal{A}\} \\ \mathcal{L}_0(\langle a, b \rangle) &= \{R \mid R(a, b) \in \mathcal{A}\} \end{aligned}$$

- UNA $\Rightarrow I_0 = \{a_i \not\doteq a_j \mid 1 \leq i < j \leq n\}$, egyébként ennek részhalmaza.
- A kezdeti $\mathbf{T}_0$ tablóban levő a_i csomópontokat gyökérpontoknak hívjuk.
- Adatstruktúra módosítás: az I komponensben $x \doteq y$ *egyenlőség* is lehet.
- A $\leqslant$ szabály két ágra bomlik:
  - 1. eset: ha van nem-gyökérpont összevonandó: nincs változás
  - 2. eset: ha két gyökérpontot kell összevonni: a megszűntetendő y-ból induló (és y-ba érkező) éleket áthelyezzük a megmaradó z ponthoz, és felvesszük az $y \doteq z$ egyenlőséget.
- Jelölés: $\mathcal{L}^*(\langle u, v \rangle) = \begin{cases} \mathcal{L}(\langle u, v \rangle) & \text{ha } \langle u, v \rangle \in E \\ \emptyset & \text{egyébként} \end{cases}$

Az adatdobozos $\mathcal{SHIQ}$ tabló-algoritmus $\leqslant$-szabálya

Feltétel:	$(\leqslant n\, R.C) \in \mathcal{L}(x)$, x nem közvetlenül blokkolt, és x-nek van $n+1$ R-szomszédja $(y_0, \ldots, y_n)$, hogy $C \in \mathcal{L}(y_i)$, és van köztük két összevonható. Minden $(0 \leq i < j \leq n)$-re ahol y_i és y_j összevonható
$\mathbf{T}_{ij}$ ***új állapot:***	1. eset: y_i és y_j közül legalább az egyik nem gyökérpont, jelölje $\{y, z\} = \{y_i, y_j\}$ úgy, hogy $y \not\Rightarrow x$: $\mathcal{L}'(z) = \mathcal{L}(z) \cup \mathcal{L}(y)$, $\mathcal{L}'(\langle x, y \rangle) = \emptyset$, $\mathcal{L}'(\langle z, x \rangle) = \mathcal{L}(\langle z, x \rangle) \cup \mathrm{Inv}^*(\mathcal{L}(\langle x, y \rangle))$ ha $z \Rightarrow x$ $\mathcal{L}'(\langle x, z \rangle) = \mathcal{L}(\langle x, z \rangle) \cup \mathcal{L}(\langle x, y \rangle)$ ha $z \not\Rightarrow x$ $I' = I[y \to z]$ (y minden előfordulását z-re cseréljük).
$\mathbf{T}_{ij}$ ***új állapot:***	2. eset: y_i és y_j gyökérpont, jelölje $y = y_i, z = y_j$ $\mathcal{L}'(z) = \mathcal{L}(z) \cup \mathcal{L}(y)$, $V' = V \setminus \{y\}$, $E' = E \setminus \{\langle u, v \rangle \mid \langle u, v \rangle \in E, \text{ és vagy } u = y \text{ vagy } v = y\}$, $\mathcal{L}'(\langle z, u \rangle) = \mathcal{L}^*(\langle z, u \rangle) \cup \mathcal{L}(\langle y, u \rangle), \forall u$ melyre $\langle y, u \rangle \in E$, $\mathcal{L}'(\langle u, z \rangle) = \mathcal{L}^*(\langle u, z \rangle) \cup \mathcal{L}(\langle u, y \rangle) \forall u$ melyre $\langle u, y \rangle \in E$, $I' = \{y \doteq z\} \cup I[y \to z]$ (minden y-t z-re cserélünk).

A tabló-algoritmus optimalizálása (a könyv 5.4 alfejezete)

- A szabályok végrehajtásának sorrendezése: az utódcsomópontok létrehozása
- Egy nem-redundáns, ú.n. *lexikai* normálalak használata
- A T-doboz axiómák transzformációja
- Fogalmak késleltetett kibontása
- Szemantikus elágaztatás: ha $C \sqcup D$ esetén C hozzávétele nem vezetett sikerre, akkor a másik ágon D mellé $\neg C$-t is felvesszük
- Metszés: a determinisztikusan feldolgozható diszjunkciók felderítése
- Intelligens, függés által irányított visszalépés
- Heurisztikák alkalmazása a diszjunkciók kibontásakor
- Gyorsítótár alkalmazása
- Speciális módszerek alkalmazása a T-doboz osztályozási feladat megoldására

Az $\mathcal{ALCN}$ tabló Haskell megvalósítása (a könyv 6. fejezete)

- $\mathcal{ALCN}$ terminológiai következtetés, nem-üres T-dobozt is megengedve
- A külső interfész:

```
data Expr = Top                          | Exists String Expr
          | Atomic String                | All String Expr
          | Not Expr                     | AtLeast Int String
          | Expr `And` Expr              | AtMost Int String
          | Expr `Or` Expr               deriving (Eq, Ord, Show)

data Axiom = Expr `Implies` Expr

satisfiable :: Expr -> [Axiom] -> Bool
satisfiable c tbox = not (null (models c tbox))
```

- A tabló adatstruktúra megvalósítása

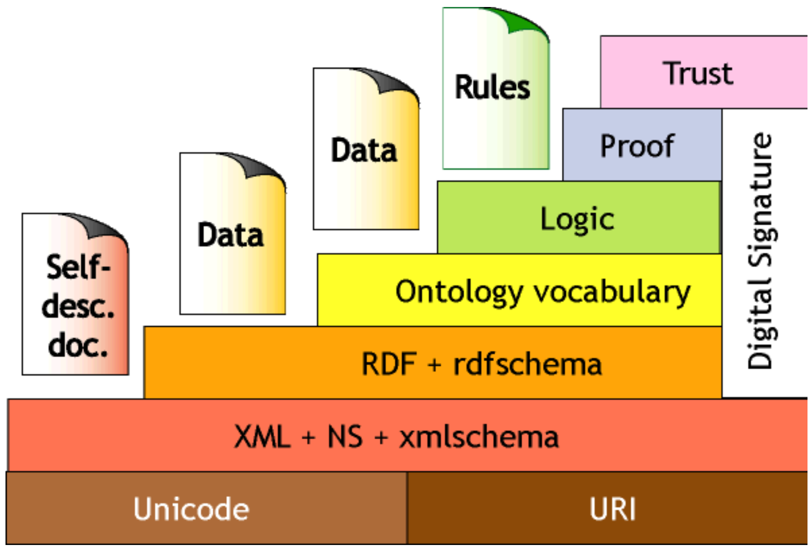
```
data Tableau=Tableau { nodes :: [Node], edges :: [Edge],
                      topExpr :: Expr, nextId :: Id, diffs :: [(Id,Id)] }

data Node = Node     { nid :: Id,      exprs :: [Expr] }

data Edge = Edge     { parentId :: Id, role :: String, childId :: Id }
```


III. rész

A szemantikus világháló

A szemantikus világháló víziója

- A szemantikus világháló rétegei (Semantic Web layer cake) – Tim Berners-Lee



Tartalom

Az XML alapfogalmai – a könyv 2.2 alfejezete[1]

- XML = Extensible Markup Language, azaz Kiterjeszthető Jelölőnyelv
  - Az XML egy metanyelv (nyelvcsalád), amelynek segítségével strukturált szöveges adatokat írhatunk le
  - XML alkalmazás: egy konkrét XML nyelv, adott szintaxissal
  - XML dokumentum: egy XML alkalmazás egy mondata
- Az XML célja
  - Egymással együttműködő rendszerek adatcsere-formátuma
  - Hatékony gép-gép kommunikáció

```
<?xml version="1.0" encoding="ISO-8859-2">
<uzenet> <kitol>Piroska</kitol>
         <kinek>Nagymama</kinek>
         <torzs>Megyek hozzád ma délután</torzs>
         <utoirat>Viszek kalácsot</utoirat>
         <utoirat>Képzeld, álítólag farkasok vannak az erdőben!
                                 </utoirat>
</uzenet>
```

[1] Ez és az ezt követő diák Zombori Zsolt prezentációi alapján készültek

Az XML dokumentumok részei

- Elemek
  - Felépítés: nyitó címke (pl. <uu>), törzs, záró címke (pl. </uu>); üres elem esetén nyitó-záró címke (pl. <uu/>)
  - Típusai: összetett, egyszerű, vegyes, üres
  - Elemek hierarchikus viszonya rögzített: egy fát határoznak meg
- Attribútumok
  - Elemek tetszőleges számú attribútummal rendelkezhetnek
  - Az attribútum értéke csak egyszerű típusú lehet (szám, literál)

```
<?xml version="1.0" encoding="ISO-8859-2">
<uzenet kitol="Piroska" kinek="Nagymama">
     <torzs>Megyek hozzád ma délután</torzs>
     <utoirat>Viszek kalácsot</utoirat>
     <utoirat>Képzeld, álítólag farkasok vannak az erdőben!
     </utoirat>
</uzenet>
```


XML névterek

- Több dokumentum esetén ütközhetnek az elem- és attribútumnevek
  - Egy XML dokumentum – a nevek egyediségét a szerző biztosítja
  - Több XML dokumentum – globális nevekre van szükség
  - A megoldás: névterek használata, az elemneveket előtaggal (prefixummal) látjuk el: <n:uzenet>...</n:uzenet>
- Univerzális (minősített) elemnevek
  - Minősített név = prefixummal kiegészített, globálisan egyedi név,
  - Az egyedi prefixum alapja az URI (Universal Resource Identifier)
  - Az URI kötött formával rendelkező literál, általában szervezethez vagy személyhez tartozik
  - xmlns → alapértelmezett névtér, xmlns:<név> → adott nevű névter.

```
<n:uzenet xmlns:n="http://www.valami.hu/" xmlns="http://bazis.hu/"
          n:kitol="Piroska" n:kinek="Nagymama">
        <n:torzs>Megyek hozzád ma délután</n:torzs>
        <n:utoirat>Viszek kalácsot</n:utoirat>
        <n:utoirat>Képzeld, álítólag farkasok vannak az erdőben!
        </n:utoirat>
</n:uzenet>
```

XML sémák – XML dokumentumok ellenőrzése

- Ha az XML formai követelményei teljesülnek: jól formázott dokumentum
- De ettől még nem biztos, hogy teljesíti a szerkezeti elvárásokat:
  - Az üzenet megfelelő gyerekelemekből áll?
  - Az üzenet feladója egy literál?
  - Szerepelhet-e két utóirat?
  - Lehet az utóirat előbb, mint az üzenet törzse?
- A megoldás: XML sémák
  - Az XML séma egy XML nyelvű dokumentum, amely egy XML alkalmazás (egy adott XML nyelv) formáját írja le
    - szintaxis
    - milyen elemek és attribútumok használhatók?
    - mi az elemek egymáshoz való viszonya?
  - Egy XML séma segítségével automatikusan meghatározható, hogy egy dokumentum megfelel-e az adott XML nyelvnek
    $\Rightarrow$ a dokumentum-feldolgozó program sokkal egyszerűbb lehet

Tartalom

RDF – Erőforrás-leíró keretrendszer – a könyv 2.3. alfejezete

- Problémák a webes kereséssel – szemantika hiánya
  - Jelentés helyett szöveges alakkal dolgozunk
  - A keresés eredménye az információ tényleges reprezentációjától (és nem a tartalmától) függ
  - Nyelvi korlátok
  - Képekhez, hangokhoz semmilyen jelentést nem tudunk társítani
  - Nem tudunk következtetni (szinonimák, taxonómiák)
- A szemantika megragadása
  - Helyezzünk el metainformációt a Weben!
    Metainformáció: információ, mely információról szól
    (link egy másik oldalról, szerző neve, stb.)
  - Az adatforrások számára tegyük lehetővé, hogy metaadatot szolgáltassanak magukról
  - A metaadat legyen egységes, strukturált, géppel feldolgozható

A Szemantikus Világháló alapja, az RDF

- A Szemantikus Világháló célkitűzése:
  - Oldalakhoz metainformáció társítása
  - Következtetéshez szükséges háttértudás leírása
  - Mindezt egységesen és automatikusan feldolgozható módon
- Metainformáció társítása
  - Tetszőleges webes „erőforrás”
  - Tetszőleges mondanivaló
  - Nagyon általános keretrendszer kell
  - RDF: Resource Description Framework
- RDF
  - Az RDF segítségével erőforrásokról tehetünk kijelentéseket
  - Erőforrás bármi lehet
  - Az a lényeg, hogy egyértelműen azonosítható legyen

Erőforrások

- Erőforrásokra egyértelmű azonosítóval hivatkozunk (például URL)
- Általánosabban Universal Resource Identifier (URI), példák:

```
http://www.cs.uwyo.edu/index.html
mailto:zombori@cs.bme.hu
file:///c:/examples/cat.rdf
uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C1C14882
```

- Az URI-k fajtái
  - Abszolút URI: teljes, egyértelműen azonosít
  - Relatív URI: adott környezetben azonosít, azon kívül csak egy bázis URI-val együtt
  - Bázis segítségével feloldjuk a relatív URI-t és abszolút URI-t kapunk
  - Komplex honlap részei könnyen tudnak egymásra hivatkozni
- Az URI-k jellemzői
  - Ugyanarról az erőforrásról több különböző helyen is tehetünk kijelentéseket
  - Bárki bármit mondhat – csak a megfelelő URI kell hozzá
  - Más helyről származó információtöredékek kombinálhatóak

RDF állítások

- RDF-ben erőforrások kapcsolatrendszerét tudjuk leírni
- Egy RDF leírás kijelentések, ún, hármasok halmaza:

(Erőforrás1, Kapcsolat, Erőforrás2-vagy-literál)

```
(www.cs.bme.hu, tulajdonosa, SZIT)
(SZIT, típusa, Tanszék)
(SZIT, vezetője, Katona Gyula)
```

- Egy RDF leírás megfeleltethető egy gráfnak

```
              www.cs.bme.hu
                    |
                    | tulajdonosa
                   \/
                  SZIT
                  / \
     típusa      /   \   vezetője
                /     \
              \/       \/
           Tanszék   Katona Gyula
```


Az RDF fő jellemzői

- Az RDF hármasok építőkövei:
  - Erőforrás: bármi aminek URI-ja van
  - Tulajdonság: speciális, kapcsolatot jelölő erőforrás, pl, `vezetője`
    Fontos: bizonyos tulajdonságok jelentését az RDF (séma) lerögzíti
  - Literál: karaktersorozat
- Egy RDF kijelentés: egy *(alany, állítmány, tárgy)* alakú hármas
  - az alany: erőforrás
  - az állítmány: tulajdonság (ez is erőforrás)
  - a tárgy: erőforrás vagy literál
- RDF leírás: kijelentések halmaza (sorrend nem számít)
  - Az RDF leírás jelentése: a kijelentések igazak
  - RDF segítségével bináris relációkat írhatunk le
- Az RDF szintaxis – az adatmodell nem rögzíti a formátumot
  Három adatmodell reprezentáció:
  - Hármasok halmaza
  - Címkézett, irányított gráf
  - XML formátum

Az RDF gráf

- Az RDF gráf elemei
  - Csomópont: erőforrás vagy literál
  - Él: tulajdonság (URI-val ellátott), csak abszolút URI lehet
  - Tulajdonságról is lehet állítást megfogalmazni
- Példa: a Magányos Cédrus festője Csontváry Kosztka Tivadar.
  - `([http://.../cedrus.html], festője, "Cs.  K. Tivadar")`
  - Gráf alak:

```
                                 festője
[http://.../cedrus.html]  --------------->  "Cs. K. Tivadar"
```

  - Modellezzük azt is, hogy Csontváry 1853-ban született!
    - Literál nem lehet alany. Ún. köztes erőforrás használunk:

```
                                                 neve
                           festője +----+ ------> "Cs. K. T."
[http://.../cedrus.html]   ------> |    | sz.éve
                                   +----+ ------>  1853
```

  - Köztes erőforrások – nem hivatkozhatók, nincs URI-juk
  - Az információ strukturáltságát növelik
  - Összetett adatok modellezése, pl. cím = (város, utca, házszám)

Az RDF XML szintaxisa

- Az RDF gráf linearizálása
- Bizonyos XML elemek speciális jelentéssel bírnak
- Alkalmazások közti adatcserére alkalmas
- Példa: Kis Ádám, aki ember, email cime kis@cs.bme.hu.

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:s="http://www.utils.org/utils#">

<rdf:Description rdf:about=http://cs.bme.hu/kis/#about>
       <s:neve>Kis Ádám</s:neve>
       <s:levélcíme rdf:resource=mailto:kis@cs.bme.hu/>
       <rdf:type rdf:resource=
                http://www.thing.org/rdf/schemas/simple#Ember/>
 </rdf:Description>

</rdf:RDF>
```

- Az `rdf:type` speciális tulajdonság:
  adott osztályba (fogalomba) tartozást ír le

Az RDF XML szintaxisa – részletek

```
<rdf:Description rdf:about=http://cs.bme.hu/kis/#about>
       <s:neve>Kis Ádám</s:neve>
       <s:levélcíme rdf:resource=mailto:kis@cs.bme.hu/>
       <rdf:type rdf:resource=
                http://www.thing.org/rdf/schemas/simple#Ember/>
 </rdf:Description>
```

- Megosztott alany használata: a fenti példa három RDF állítást ír le, mindegyiknek az alanya `http://cs.bme.hu/kis/#about`
- A tulajdonság is erőforrás – URI

  ```
  xmlns:s="http://www.utils.org/utils#"
  ...
  <s:neve>Kis Ádám</s:neve>
  ```

  A hivatkozott tulajdonság URI-ja `http://www.utils.org/utils#neve`
- Erőforrást tárgypozícióban csak a `rdf:resource` attribútum segítségével adhatunk meg, az alábbi példa hibás:

  ```
  <s:levélcíme>mailto:kis@cs.bme.hu</s:levélcíme>
  ```


Az RDF XML szintaxisa – részletek (2)

```
<rdf:Description rdf:about=http://cs.bme.hu/kis/#about>
       <s:neve>Kis Ádám</s:neve>
       <s:levélcíme rdf:resource=mailto:kis@cs.bme.hu/>
       <rdf:type rdf:resource=
                http://www.thing.org/rdf/schemas/simple#Ember/>
 </rdf:Description>
```

- Típusmegadás egyszerűbb szintaxissal – a fenti helyett írható:

  ```
  xmlns:ss="http://www.thing.org/rdf/schemas/simple#"
  ...
  <ss:Ember rdf:about=http://cs.bme.hu/kis/#about>
         <s:neve>Kis Ádám</s:neve>
         <s:levélcíme rdf:resource=mailto:kis@cs.bme.hu/>
   </ss:Ember>
  ```


Az RDF XML szintaxisa – `parseType` attribútum

- Az `rdf:parseType` attribútum segítségével egy tárgypozícióban levő elem interpretációját lehet megváltoztatni

  ```
  <rdf:Description rdf:about="http://192.168.121.8">
        <dc:Title rdf:parseType="Literal">
                Ez az <I>én</I> gépem!
        </dc:Title>
        <dc:Creator>Compaq</dl:Creator>
  </rdf:Description>
  ```
- A fenti példában az „`Ez az <I>én</I> gépem!`” szöveg literálként értelmeződik, annak ellenére, hogy XML elem van benne
- Az `rdf:parseType` használata köztes erőforrások leírására

  ```
  <rdf:Description
   rdf:about="http://.../cedrus.htm">
   <s:festője rdf:parseType="Resource">
    <s:neve>Csontváry Kosztka Tivadar</s:neve>
    <s:születésiÉve>1853</s:születésiÉve>
   </s:festője>
  </rdf:Description>
  ```

Az RDF XML szintaxisa – rdf:nodeID attribútum

- Az `rdf:nodeID` attribútummal egy lokális azonosítót vezethetünk be, amit egy RDF leíráson belül többször is használhatunk, pl. köztes erőforrások megnevezésére

```
<rdf:Description rdf:about="http://.../cedrus.htm">
      <s:festője rdf:nodeID="lokális_azonosító1"/>
</rdf:Description>

<rdf:Description rdf:nodeID="lokális_azonosító1">
      <s:neve>Csontváry Kosztka Tivadar</s:neve>
      <s:születésiÉve>1853</s:születésiÉve>
</rdf:Description>
```


Az RDF XML szintaxisa – rdf:ID attribútum

- Új URI bevezetésére használható az `rdf:ID` attribútum
- Egy azonosító csak egyszer szerepelhet

```
<rdf:Description rdf:ID="munkatárs1">
      <s:neve>Szép Hajnalka</s:neve>
      <s:fizetése>220</s:fizetése>
</rdf:Description>
```

- Abszolút URI képzése: bázis URI + # + ID: `bazis.hu/bazis.html#munkatárs1`

RDF további fontos témák – áttekintés

- Nem bináris relációk kezelése:
  Köztes erőforrás bevezetésével több bináris relációra bontjuk
  - Példa: „A és B mérkőzésének eredménye E” modellezése
  - Bevezetünk egy köztes erőforrást, legyen ez M (mérkőzés)
  - Három új tulajdonság: házigazdája, vendégcsapata, eredménye
  - Modellezés: M házigazdája A, M vendégcsapata B, M eredménye E
- Magasabbrendű kijelentések
  - A `rdf:Statement` osztály példányai egy RDF állítást neveznek meg.

```
<rdf:Description nodeID="azonosito1">
 <rdf:type
      rdf:resource=http://www.w3.org/...#Statement/>
 <rdf:subject
  rdf:resource=http://festok.hu#csontvary/>
 <rdf:predicate rdf:resource=http://www.u.org#szulinap/>
 <rdf:object>1755 </rdf:object>
</rdf:Description>
```


RDF – további fontos témák áttekintése (2)

- Konténerek, kollekciók
  - Egy csoportra vonatkozó állítások modellezésére használhatók
  - Ami a csoportra igaz, egyedeire nem feltétlenül igaz!
  - Nyílt végű konténerek `rdf:Bag`, `rdf:Seq`, `rdf:Alt`.
    Zárt végű kollekció: `rdf:List`
  - `rdf:Bag` – Sorrend nem számít, egy elem többször is előfordulhat
  - `rdf:Seq` – Rendezett, sorrend számít
  - `rdf:Alt` – Az elemek lehetséges alternatívákat jelölnek.
    Legalább egyelemű, az első elem az alapértelmezett
  - `rdf:List` – Zárt végű skatulyázott lista
- Típusos literálok
  - Az RDF nem ismer beépített típusokat, de definiál egy `rdf:datatype` attribútumot
  - Az RDF az XML séma által definiált típusok használatát ajánlja

```
<s:szulinap rdf:dataType="http://www.w3.org/2001/XMLSchema#date">
                  1853-07-05
</s:szulinap>
```

Tartalom

Az RDF Séma – a könyv 2.4. alfejezete

- Az RDF sémával leírható bizonyos fajta egyszerű háttértudás, pl. osztályok (fogalmak) ill. tulajdonságok (szerepek) hierarchiája
- Példa: Anna barátja Éva. Igaz-e, hogy Anna ismerőse Éva?
- Igen, feltéve, hogy ismerjük a „barátja” és „ismerőse” szavak jelentését
- A leíró logikai barátja $\sqsubseteq$ ismerőse axióma RDFS-ben:

```
<rdf:RDF
        xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
        xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#>

<rdf:Description rdf:ID="ismerőse">
        <rdf:type rdf:resource=rdf:Property/>
</rdf:Description>

<rdf:Property rdf:ID="barátja">
        <rdfs:subPropertyOf rdf:resource="#ismerőse">
</rdf:Property>
</rdf:RDF>
```


Az RDF Séma – osztályhierarchia

- Marad az RDF szintaxis
- Osztályokat és tulajdonságokat definiálhatunk
- Leírhatjuk az osztályok és tulajdonságok hierarchikus viszonyát
- Sémaleírás ill. adatleírás (T-doboz ill. A-doboz) – más-más metaszint
- Mindkettő leírására ugyanazt az RDF szintaxist használjuk
- Osztályhierarchia megadása:

```
<rdf:Description rdf:ID="Állat">
        <rdf:type rdf:resource=rdfs:Class/>
</rdf:Description>                  Hüllő ⊑ Állat:
<rdfs:Class rdfID="Hüllő">
        <rdfs:subClassOf rdf:resource="#Állat"/>
</rdfs:Class>                       Emlős ⊑ Állat:
<rdfs:Class rdfID="Emlős">
        <rdfs:subClassOf rdf:resource="#Állat"/>
</rdfs:Class>                       Ember ⊑ Emlős:
<rdfs:Class rdfID="Ember">
        <rdfs:subClassOf rdf:resource="#Emlős"/>
</rdfs:Class>
```


Az RDF Séma – tulajdonság-hierarchia

```
<rdf:Description rdf:ID="ismerőse">
        <rdf:type rdf:resource=rdf:Property/>
</rdf:Description>

                                  barátja ⊑ ismerőse:
<rdf:Property rdf:ID="barátja">
        <rdfs:subPropertyOf rdf:resource="#ismerőse">
</rdf:Property>

                                  jóbarátja ⊑ barátja:
<rdf:Property rdf:ID="jóbarátja">
        <rdfs:subPropertyOf rdf:resource="#barátja">
</rdf:Property>
```

Az RDF Séma – tulajdonságkorlátozások

- A `leánykori neve` tulajdonság értelmezési tartománya (domain) az `Ember` és `Nőnemű` osztályok metszete:

```
<rdf:Property rdf:ID="leánykori neve">
        <rdfs:domain rdf:resource="Ember"/>
        <rdfs:domain rdf:resource="Nőnemű"/>
</rdf:Property>
```

- A `szerzője` tulajdonság értelmezési tartománya `Könyv`, értékkészlete (range) pedig `Ember`:

```
<rdf:Property rdf:about="szerzője">
        <rdfs:domain rdf:resource="Könyv"/>
        <rdfs:range rdf:resource="Ember"/>
</rdf:Property>
```


Az RDF Séma egy alkalmazása

- RDF sémák segítségével készíthető saját szótár, de ez nem célszerű
- Használjunk általánosan elfogadott, közös szótárakat!
- Magas szintű, általános fogalmak, RDFS-ben ezeket lehet bővíteni
- A Dublin Core egy 1995-ben létrehozott RDF séma
  - Cél: dokumentumok automatikus keresése fő jellemzőik szerint
  - Eredmény: 15 szabványos elem (tulajdonság) megnevezése: Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights

```
<rdf:RDF xmlns:rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#
         xmlns:dc=http://purl.org/dc/elements/1.1/>
   <rdf:Description rdf:about=http://wald.hu/ikon.mp3>
           <dc:creator>Kovács Ákos</dc:creator>
           <dc:title>Ikon</dc:title>
           <dc:description> Az Ikon című album címadó dala.
           </dc:description>
           <dc:date>2003-04-21</dc:date>
   </rdf:Description>
</rdf:RDF>
```


Tartalom

Az OWL nyelv – a könyv 7. fejezete

- Az RDF és RDFS nagyon egyszerű ontológialeíró nyelvek
- Az OWL ezek leíró logikai kiterjesztése
- Az OWL nyelv tekinthető a LL egy konkrét szintaxisának
- Miért pont ilyen lett a világháló ontológia nyelve?
  Ugyanaz motiválta, mint az RDF-et
  - XML alapú
  - Weben kényelmesen elhelyezhető
  - A jelenlegi webes keresők támogatják az XML dokumentumok feldolgozását
  - Adatcsere formátum alkalmazások között
  - LL háttér biztosítja a következtetést
- Mi az OWL?
  - Egy OWL dokumentum egy érvényes RDF leírás
  - OWL bevezet egy erőforrás-halmazt és rögzíti a jelentését
  - Ugyanúgy, mint az RDFS

A „lányos apa” példa OWL nyelven

```
<rdf:RDF xml:base=http://ww.cs.bme.hu/vima9000# xmlns:rdf=...>

<owl:Class rdf:ID="Ember" />
<owl:Class rdf:ID="Nő">   <rdfs:subClassOf rdf:resource="#Ember"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="gyereke">
  <rdfs:domain rdf:resource="#Ember"/>
  <rdfs:range rdf:resource="#Ember"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="LányosApa">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about:"#Ember"/>
    <owl:Class> <owl:complementOf rdf:resource="#Nő"/> </owl:Class>
    <owl:Restriction> <owl:onProperty rdf:resource="#gyereke" />
            <owl:allValuesFrom rdf:resource="#Nő" />
    </owl:Restriction>
  </owl:interectionOf>
</owl:Class>
</rdf:RDF>
```


Az OWL1 résznyelvei

- OWL Full
  - Minden RDF konstrukció használható
  - Nem ágyazható semmilyen DL nyelvbe
  - Probléma: magasabbrendű kijelentések
- OWL DL
  - Közvetlenül fordítható DL-re $\rightarrow \mathcal{SHOIN}(\mathbf{D})$
  - Erőforrásoknak meghatározott típusa van:
    - Egyed, osztály, absztrakt tulajdonság,
    - konkrét érték, konkrét osztály, konkrét tulajdonság
- OWL Lite
  - Leegyszerűsített OWL DL
  - Megfelel a $\mathcal{SHIF}(\mathbf{D})$ nyelvnek
  - Átmenet az RDFS és az OWL DL között
  - Nagyon hatékony következtetés

OWL osztályok

- Osztályok fajtái (megfelelnek a fogalomkifejezések fajtáinak)
  - Elnevezett osztály
  - Enumerációs osztály
  - Tulajdonságkorlátozásos osztály
  - Metszetosztály
  - Unióosztály
  - Komplementer osztály
- Elnevezett osztály
  - DL megfelelője: atomi fogalom
  - Beépített elnevezett osztályok: `owl:Thing` ($\top$) és `owl:Nothing` ($\bot$)
  - Példa: `<owl:Class rdf:ID="Ember"/>`

OWL osztályok

- Enumerációs osztály, DL megfelelője: nominálisok uniója
  - Nem megengedett OWL Lite-ban

```
<owl:Class>
        <owl:oneof rdf:parseType="Collection">
                <owl:Thing rdf:about="#Hétfő"/>
                <owl:Thing rdf:about="#Kedd"/>              ...
        </owl:oneof>
</owl:Class>
```

- Tulajdonságkorlátozásos osztály
  - Értékkorlátozás
  - Számosságkorlátozás
- Egy `P` tulajdonságra (szerepre) vonatkozó korlátozás formája (egyszerre többfajta korlátozás is megadható):

```
<owl:Restriction>
        <owl:onProperty rdf:resource="P"/>
        korlátozások
</owl:Restriction>
```

OWL osztályok

- Értékkorlátozások, DL megfelelőik: $(\forall R.C), (\exists R.C), (\exists R.\{a\})$ (nominálisra vonatkozó egzisztenciális kvantor)

```
<owl:Restriction>
    <owl:onProperty rdf:resource="#gyereke"/>
    <owl:allValuesFrom rdf:resource="#Szőke"/>
    <owl:someValuesFrom rdf:resource="#Nő"/>
    <owl:hasValue rdf:resource="#Judit"/>
</owl:Restriction>
```

- Számosságkorlátozások, DL megfelelőik: $(\leqslant n\, R), (\geqslant n\, R), (= n\, R)$

```
<owl:Restriction>
    <owl:onProperty rdf:resource="#alkalmazottja"/>
    <owl:minCardinality rdf:datatype="&xsd;nonnegativeInteger">
            3
    </owl:minCardinality>
    <owl:maxCardinality rdf:datatype="&xsd;nonnegativeInteger">
            50
    </owl:maxCardinality>
</owl:Restriction>
```


OWL osztályok

- Metszetosztály

```
<owl:Class> <owl:intersectionOf rdf:parseType="Collection">
                <owl:Class rdf:resource="#Magas"/>
                <owl:Class rdf:resource="#Barna"/>
            </owl:intersectionOf> </owl:Class>
```

- Unióosztály

```
<owl:Class> <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="#Nő"/>
                <owl:Restriction>
                  <owl:onProperty rdf:resource="#hajszíne"/>
                  <owl:hasValue>Barna</owl:hasValue>
                </owl:Restriction>
            </owl:unionOf> </owl:Class>
```

- Komplementer osztály

```
<owl:Class> <owl:complementOf>
                <owl:Class rdf:about="#Szőke"/>
        </owl:complementOf> </owl:Class>
```


OWL axiómák

- Osztályokra vonatkozó (fogalmi) állítások
  - Fogalomtartalmazási axiómák: `rdfs:subClassOf`
  - Fogalomazonossági axiómák: `owl:equivalentClass`
  - Diszjunktsági axiómák: `owl:disjointWith`
- Fogalomtartalmazási axiómák

```
<owl:Class rdf:about="#Ház">
 <rdfs:subClassOf>
  <owl:Restriction>
     <owl:onProperty rdf:resource="#ablaka"/>
     <owl:minCardinality rdf:datatype="&xsd;nonnegativeInteger">
      3
     </owl:minCardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```


OWL axiómák

- Fogalomazonossági axiómák

```
<owl:Class rdf:about="#Hétvége">
        <owl:equivalentClass>
                <owl:Class>
                  <owl:oneof rdf:parseType="Collection">
                        <owl:Thing rdf:about="#szombat"/>
                        <owl:Thing rdf:about="#vasárnap"/>
                  </owl:oneof>
                </owl:Class>
        </owl:equivalentClass>
</owl:Class>
```

- Diszjunkció

```
<owl:Class rdf:about="#Férfi">
        <owl:disjointWith rdf:resource="#Nő"/>
</owl:Class>
```

OWL szerepek

- Nincsenek szerepkonstruktorok
- Kijelenthetjük, hogy egy szerep absztrakt vagy konkrét:

```
<owl:ObjectProperty rdf:ID="gyereke"/>
<owl:DatatypeProperty rdf:ID="mérete"/>
```

- Szerepállítások – RDF séma alapú lehetőségek

```
<owl:ObjectProperty rdf:ID="apja">
        <rdfs:subPropertyOf rdf:resource="#szülője"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="kedvencItala">
        <rdfs:domain rdf:resource="#Ember"/>
        <rdfs:range rdf:resource="#Ital"/>
</owl:ObjectProperty>
```

- Szerepállítások – szerepazonosság, inverz szerepek

```
<owl:ObjectProperty rdf:about="#gyereke">
        <owl:equivalentProperty rdf:resource="#kölyke"/>
        <owl:inverseOf rdf:resource="#szülője"/>
</owl:ObjectProperty>
```


OWL szerepek

- Szerepállítások – funkcionális, inverz funkcionális szerep

```
<owl:ObjectProperty rdf:ID="felesége">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/
</owl:ObjectProperty>
```

- Szerepállítások – tranzitivitás, szimmetria

```
<owl:SymmetricProperty rdf:ID="testvére"/>

<owl:ObjectProperty rdf:ID="része">
        <rdf:type rdf:resource="#owl;TransitiveProperty/>
</owl:ObjectProperty>
```


OWL egyedek

- Nincs UNA
- Fontos, hogy egyedekről kijelenthessük, hogy azonosak ill. különbözőek
- OWL egyedek azonossága

```
<rdf:Description rdf:about="#Rudi">
        <owl:sameAs rdf:resource="#Rudolf"/>
</rdf:Description>
```

- OWL egyedek különbözősége

```
<t:Film rdf:ID="Ötödik_Elem">
        <owl:differentFrom rdf:resource="Ponyvaregény"/>
</t:Film>

<owl:AllDifferent>
        <owl:distinctMembers rdf:parseType="Collection">
                <t:Film rdf:about="#Ötödik_Elem"/>
                <t:Film rdf:about="#Ponyvaregény"/>
                <t:Film rdf:about="#Kill_Bill"/>
        </owl:distinctMembers>
</owl:Alldifferent>
```


Az OWL 2 nyelv – újdonságok

- Áttekintés
  - Kisebb kiterjesztések, szintaktikus édesítőszerek
  - Nyelvi kiterjesztés – $\mathcal{SROIQ}(\mathbf{D})$
  - Kiterjesztett konkrét adattípusok
  - Metamodellezés
- Kisebb kiterjesztések, szintaktikus édesítőszerek
- `DisjointUnion` – diszjunkt unióból előálló osztály
- `DisjointClasses` – megadott osztályok diszjunktak
- `NegativeObjectPropertyAssertion` – $\neg$gyereke(a, b)
- `NegativeDataPropertyAssertion` – $\neg$merete$(a, 42)$

Az OWL 2 nyelv – $\mathcal{SROIQ}(\mathbf{D})$ nyelvre való kiterjesztés

- Önkorlátozás – $\exists R.\mathrm{Self}$
- Minősített számosságkorlátozás – $(\leqslant n\,R.C), (\geqslant n\,R.C), (= n\,R.C)$
- Reflexív szerep – $\forall x R(x,x)$
- Irreflexív szerep – $\forall x \neg R(x,x)$
- Antiszimmetrikus szerep – $\forall x, y(R(x,y) \rightarrow \neg R(y,x))$
- Diszjunkt szerepek – $R(x,y) \rightarrow \neg S(x,y)$
- Komplex szerephierarchia – $R1 \circ R2 \circ \ldots \circ Rn \sqsubseteq R$
- Kulcsok
  - `hasKey(Diák, neptunkódja)` –
    Minden diákot azonosít a neptun kódja.
  - `hasKey(Verseny, sportága, ideje, helye)` –
    Minden versenyt azonosít a sportág, idő, hely hármas.

Az OWL 2 nyelv

- Kiterjesztett konkrét adattípusok
  - OWL-ben csak integer és string adattípusok támogatottak
  - OWL 2-ben új adattípusok (pl. double, float, decimal)
  - OWL 2-ben lehetőség van felhasználói adattípusok definiálására, pl:
    - 18-nál nagyobb egészek
    - 18-nál kisebb, vagy 32-nél nagyobb egészek
    - legalább 3 hosszú stringek
- Metamodellezés
  - OWL-ben az erőforrásoknak jól meghatározott típusa van
  - OWL 2-ben egy erőforrás lehet egyszer egyed, egyszer osztály
    - Sas: sasok halmaza
    - Sas: egyed, mely egy fajt azonosít
  - Konkrét egyedek és osztályok, valamint tulajdonságok továbbra is csak egy szerepben fordulhatnak elő