

Ontológiák, 2.

Leíró logikák - Tabló módszer

Tabló módszer

Döntési eljárás a kielégíthetőség eldöntésére

Ha egy állítás kielégíthető, az eljárás konstruktív módon előállítja az állítás modelljét.

Ötlet: modell inkrementális építése állítással vezérelve, az állítást fentről/lefelé módon dekomponálva. Az eljárás kimerítő, ha a modellt nem találja, akkor belátható, hogy a modell nem létezik és az állítás kielégíthetetlen.

1. A Σ elmélet szintaktikai transzformációja *S kényszer rendszerbe (tabló)*.
A Σ minden állítása $\rightarrow$ egy kényszer S-ben.
2. S kényszerrendszer bővítése: *kiegészítő szabályok*.
Kiegészítő szabályok: determinisztikus bővítés
non determinisztikus bővítés – alternatívák (*elágazások*)
3. Kiegészítő szabályok alkalmazása:
ellentmondás (*ütközés*) minden ágban, vagy
egy *kiegészített ágban kifogyunk* az alkalmazható szabályokból.
4. Kiegészített kényszer rendszer modellt ad S részére (valamelyik kiegészített ág).

Egy FOL példa

$$\frac{\phi \wedge \psi}{\phi}$$

$$\frac{\phi \vee \psi}{\phi \mid \psi}$$

$$\frac{\forall x. \phi}{\phi\{x/t\}}$$

$$\frac{\exists x. \phi}{\phi\{x/Z\}}$$

$$\psi$$

$$\exists y. (p(y) \wedge \neg q(y)) \wedge \forall z. (p(z) \vee q(z))$$

$$\exists y. (p(y) \wedge \neg q(y)), \forall z. (p(z) \vee q(z))$$

$$p(\tilde{y}) \wedge \neg q(\tilde{y}), \forall z. (p(z) \vee q(z))$$

$$p(\tilde{y}), \neg q(\tilde{y}), \forall z. (p(z) \vee q(z))$$

$$p(\tilde{y}), \neg q(\tilde{y}), p(\tilde{y}) \vee q(\tilde{y})$$

$$p(\tilde{y}), \neg q(\tilde{y}), p(\tilde{y})$$

$$p(\tilde{y}), \neg q(\tilde{y}), q(\tilde{y})$$

< KIEGÉSZITETT >

< ÜTKÖZÉS >

Az állítás kielégíthető.

A kiszámított modell: $\Delta^I = \{\tilde{y}\}$, $p^I = \{\tilde{y}\}$, $q^I = \emptyset$.

Negált Normál Forma

Kiegészítő szabályok csak akkor működnek, ha az állítást a **negált normál formára hoztuk**, azaz minden negálás az atomokig be lett tolva.

Hasonló módon negált normál formára alakíthatjuk át egy *ALC* állítást, amiben a negálás az atomi fogalmak előtt helyezkedik közvetlenül:

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

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

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

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

$$\neg(\leq nR.C) \Leftrightarrow (\geq n+1R.\neg C)$$

$$\neg(\geq 0R.C) \Leftrightarrow \perp$$

$$\neg(\geq nR.C) \Leftrightarrow (\leq n-1R.\neg C), \text{ ha } n > 0$$

Tabló módszer

Legyen egy példa: $(\exists FIA.Okos) \sqcap (\exists FIA.Szép) ? \models \exists FIA.(Okos \sqcap Szép)$

azaz kielégíthető-e: $(\exists FIA.Okos) \sqcap (\exists FIA.Szép) \sqcap \neg(\exists FIA.(Okos \sqcap Szép))$

($A \models B \Leftrightarrow A \sqcap \neg B$ nem kielégíthető)

NNF: $C_0 = (\exists FIA.Okos) \sqcap (\exists FIA.Szép) \sqcap \forall FIA.(\neg Okos \sqcup \neg Szép)$

A cél egy olyan véges I interpretációt megtalálni, hogy $C^I \neq \emptyset$.

Tételezzük fel tehát, hogy létezik egy b objektum, amelyre:

$b \in (\exists FIA.Okos)^I$, $b \in (\exists FIA.Szép)^I$, és $b \in (\forall FIA.(\neg Okos \sqcup \neg Szép))^I$.

$b \in (\exists FIA.Okos)^I$ -ből következik, hogy létezik egy másik olyan c objektum, amelyre: $(b,c) \in FIA^I$ és $c \in Okos^I$.

Ugyanígy $b \in (\exists FIA.Szép)^I$ miatt létezik egy olyan objektum d, amelyre: $(b,d) \in FIA^I$ és $d \in Szép^I$.

b kielégíti a $\forall FIA.(\neg Okos \sqcup \neg Szép)$ kifejezést. c és d FIA relációban állnak vele, figyelembe kell venni: $c \in (\neg Okos \sqcup \neg Szép)^I$ és $d \in (\neg Okos \sqcup \neg Szép)^I$.

Tabló módszer

$c \in (\neg \text{Okos} \sqcup \neg \text{Szép})^I$ azt jelenti, hogy $c \in (\neg \text{Okos})^I$ vagy $c \in (\neg \text{Szép})^I$.

Ha feltesszük, hogy $c \in \neg \text{Okos}^I$, akkor ez ellentmond $c \in \text{Okos}^I$ -nek.

Tehát kénytelenek leszünk $c \in (\neg \text{Szép})^I$ -t választani.

Ugyanígy $d \in (\neg \text{Okos} \sqcup \text{Szép})^I$ esetén $d \in (\neg \text{Szép})^I$ választás ellentmondáshoz vezet, ahelyett fel kell venni $d \in (\neg \text{Okos})^I$ -t.

Tovább már nincs mit lebontani, elemi (atomi) állításokat kaptunk minden objektum jellemzésére és azok közt nincs ellentmondás.

C_0 -ra kaptunk ezzel egy modellt:

$$I = \{b, c, d\}; \quad FIA^I = \{(b,c), (b,d)\}; \quad \text{Okos}^I = \{c\} \quad \text{és} \quad \text{Szép}^I = \{d\}.$$

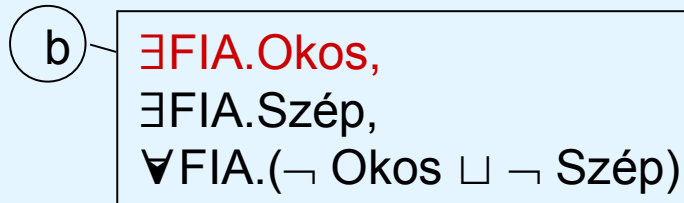
Ezzel C_0 kielégíthető, azaz

$$(\exists FIA. \text{Okos}) \sqcap (\exists FIA. \text{Szép}) \quad \text{nem részfogalma} \quad \exists FIA. (\text{Okos} \sqcap \text{Szép})\text{-nek.}$$

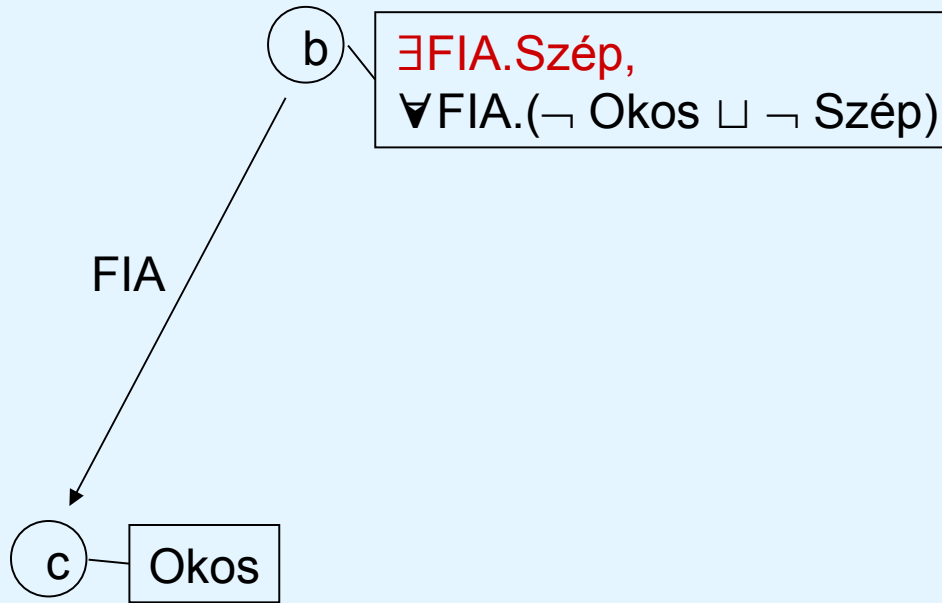
Tabló módszer

b $\exists FIA.Okos \sqcap \exists FIA.Szép \sqcap \forall FIA.(\neg Okos \sqcup \neg Szép)$

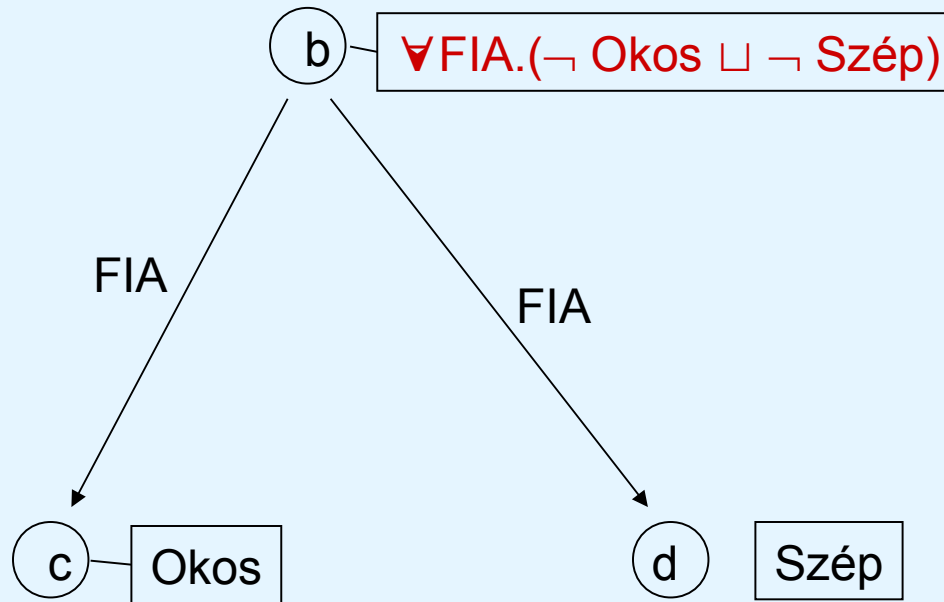
Tabló módszer



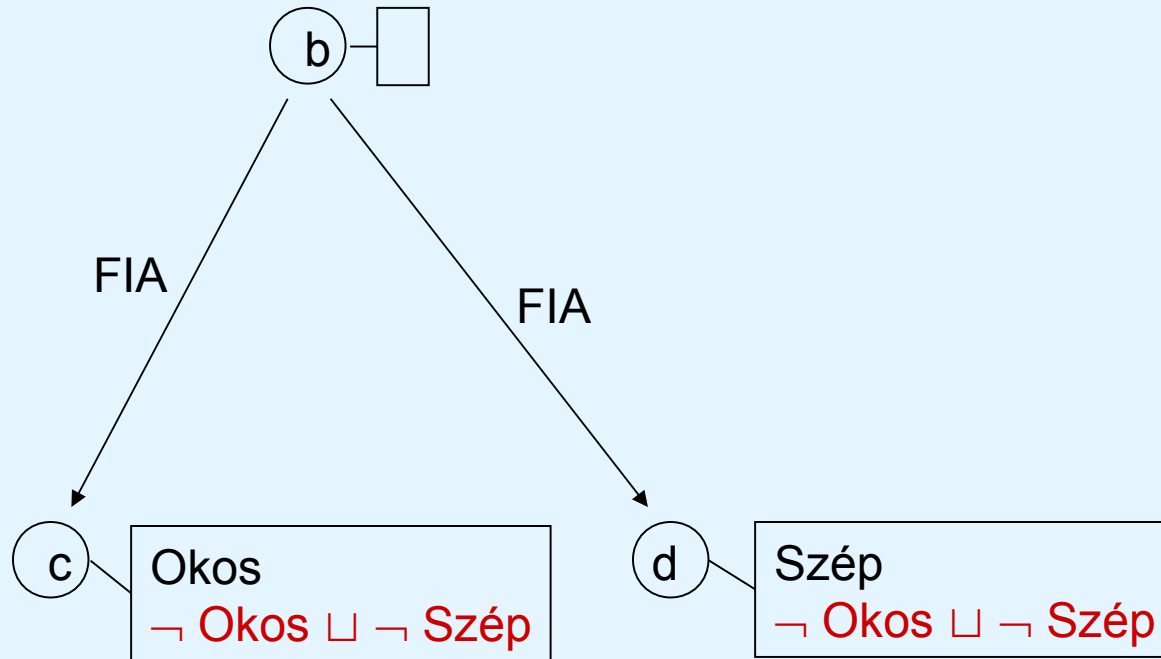
Tabló módszer



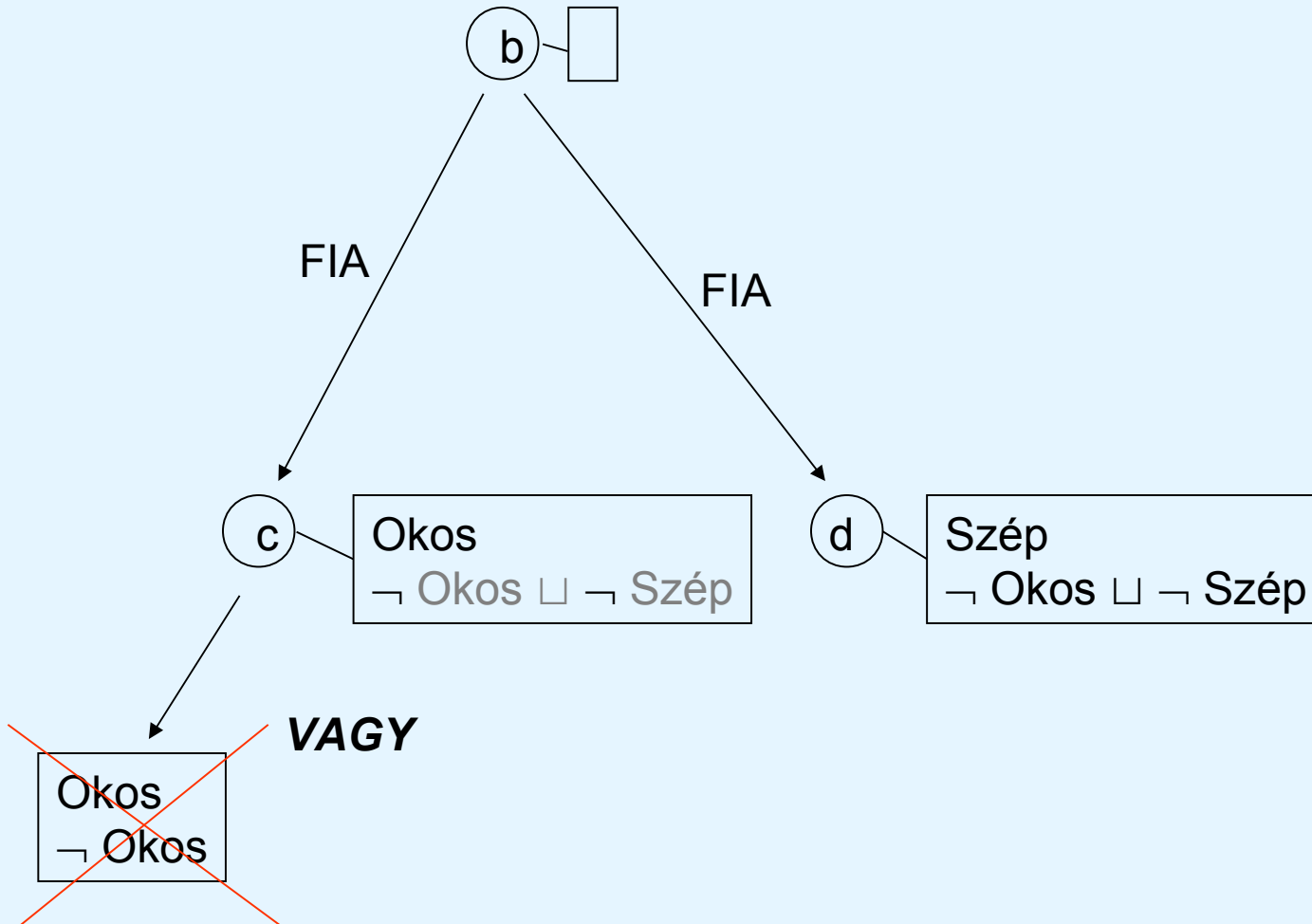
Tabló módszer



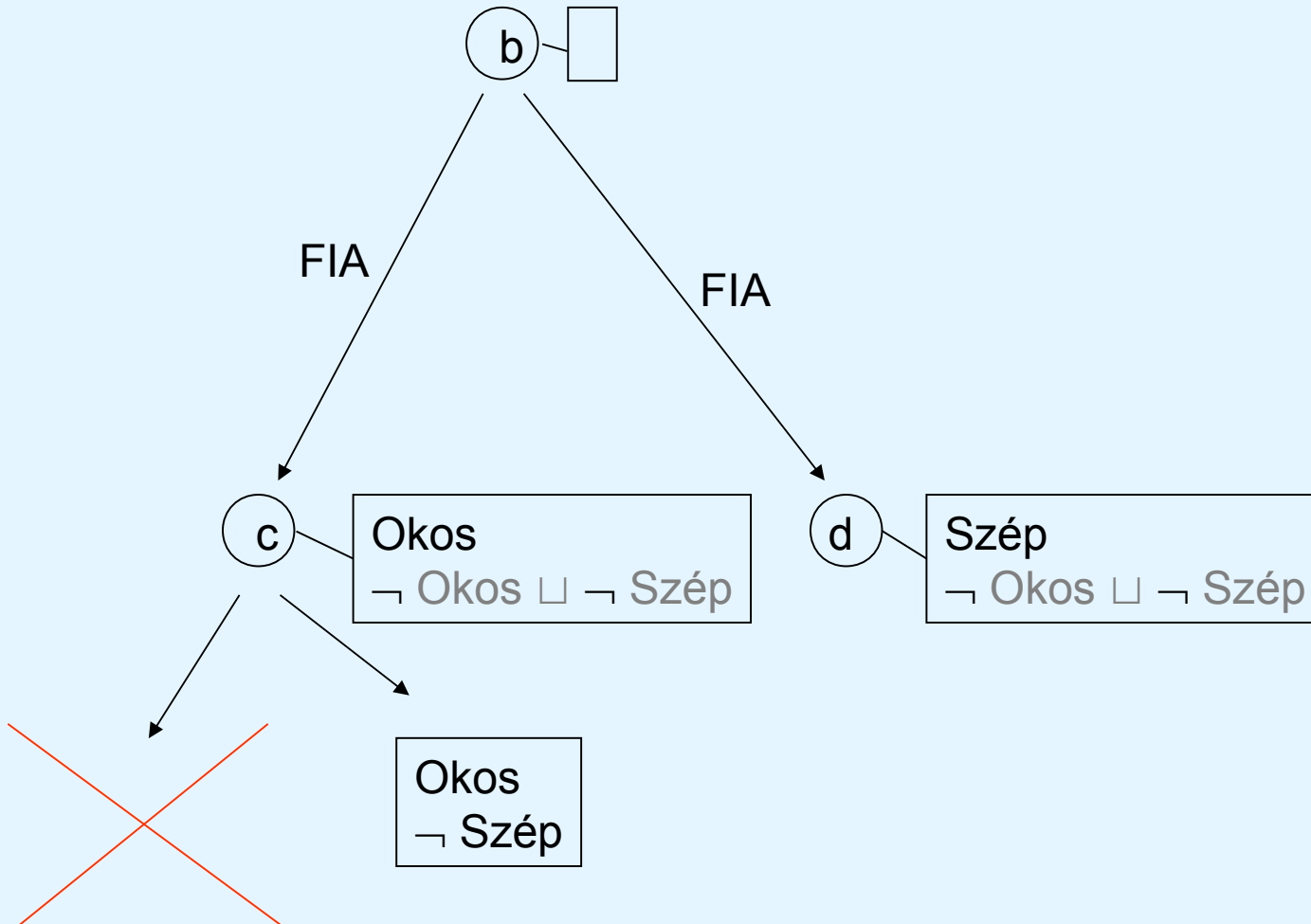
Tabló módszer



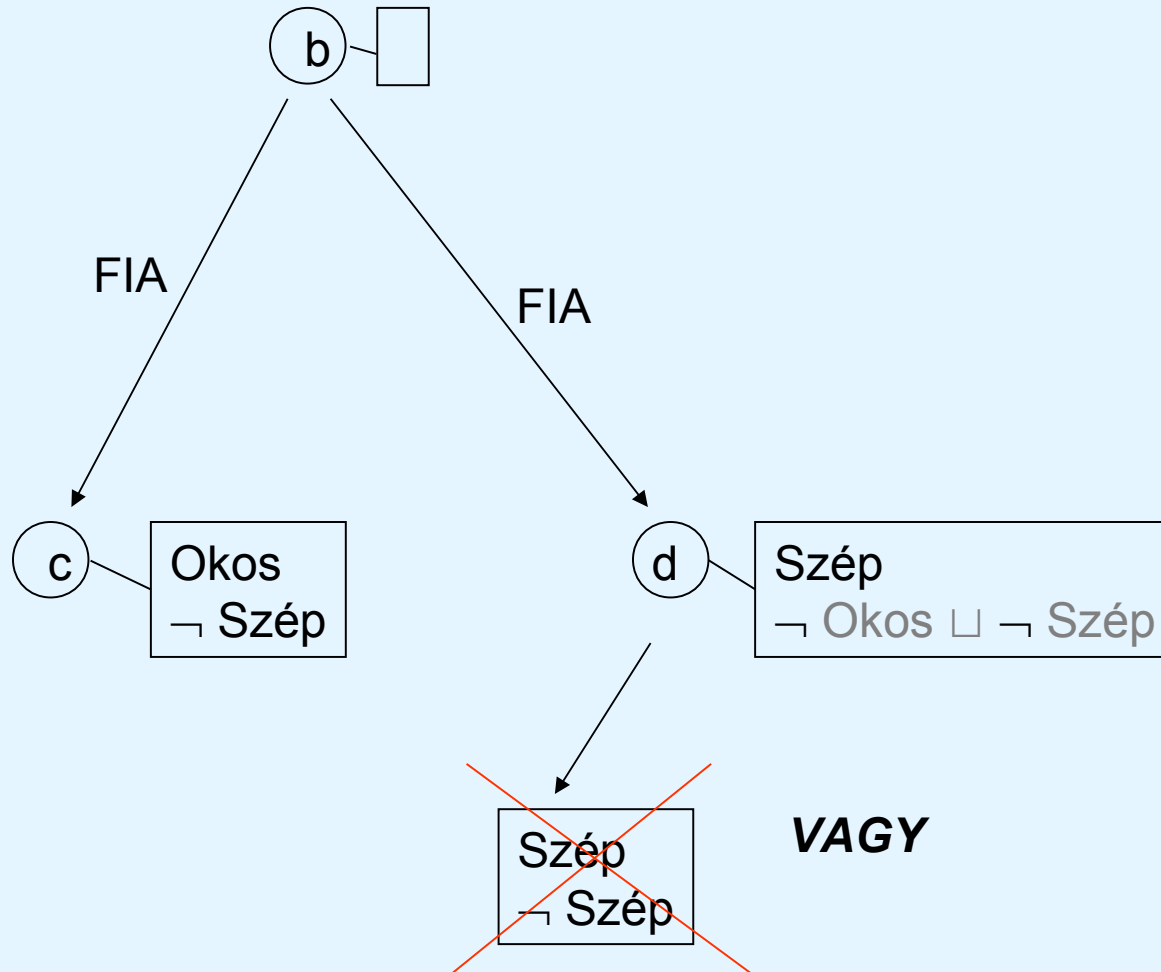
Tabló módszer



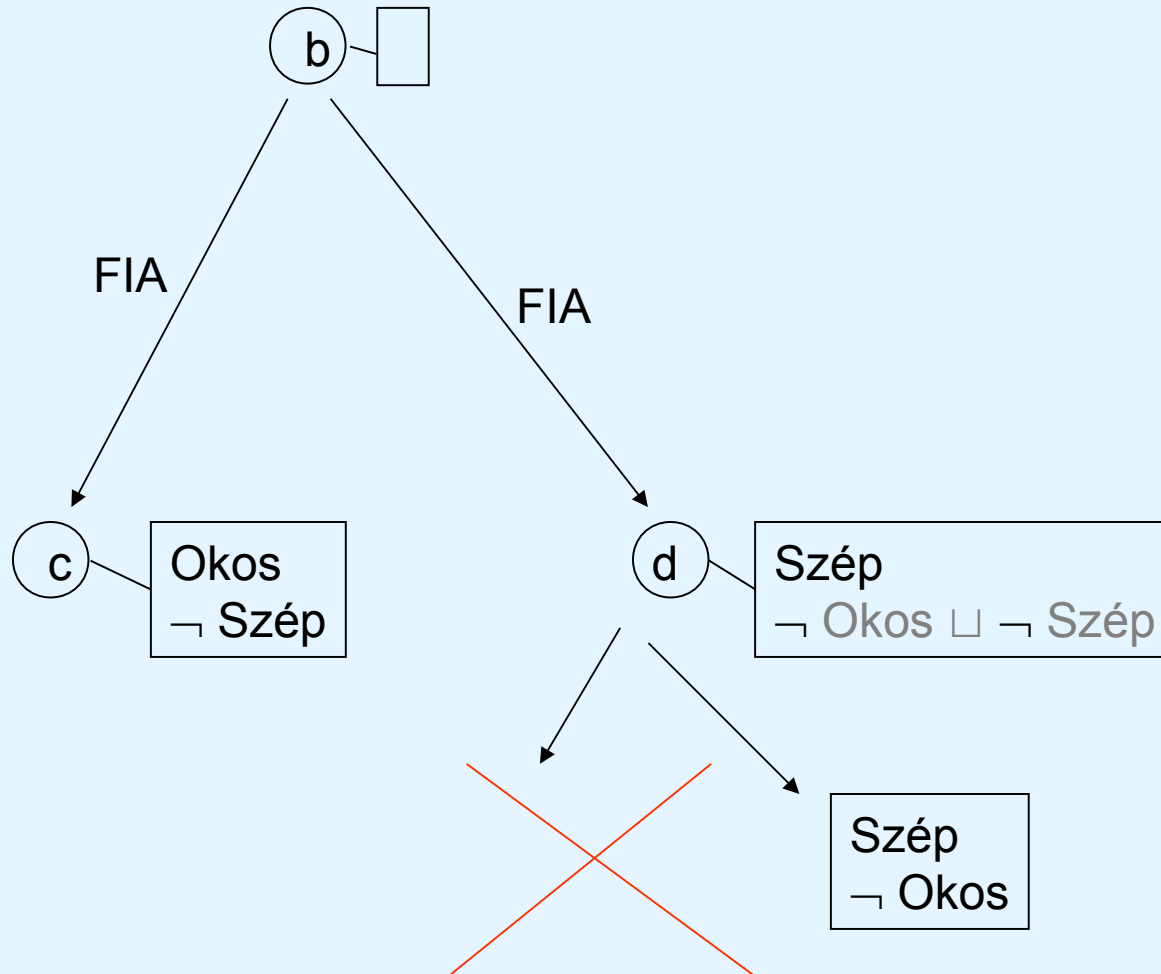
Tabló módszer



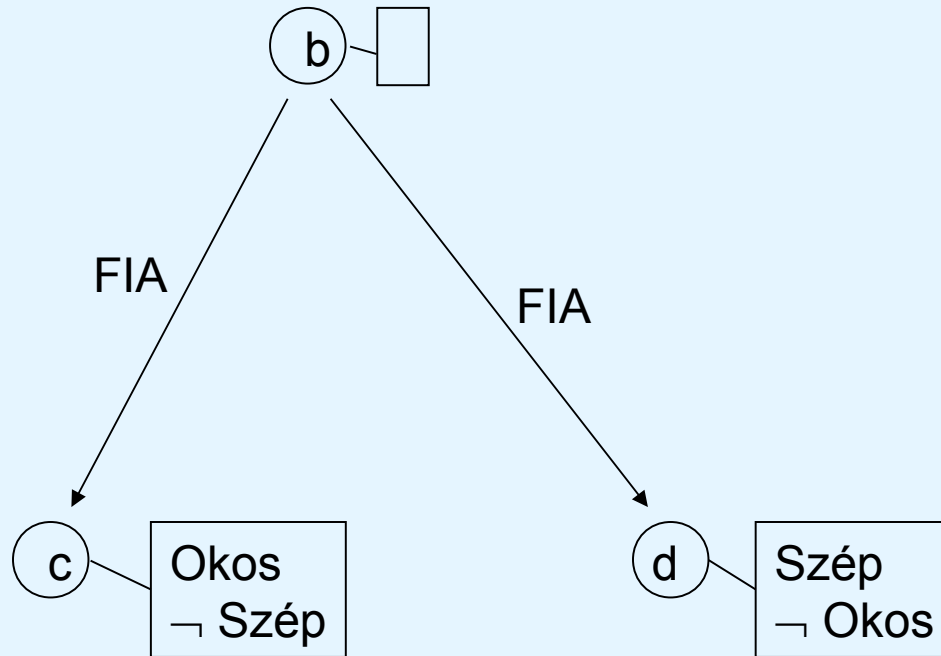
Tabló módszer



Tabló módszer



Tabló módszer



Nincs mit tovább lebontani, ellentmondás sincs.

Megszületett egy interpretáció.

Tabló módszer

Legyen a példa most:

$$(\leq 1 \text{ FIA}) \sqcap (\exists \text{ FIA. Okos}) \sqcap (\exists \text{ FIA. Szép}) \text{ ?} \sqsubseteq \exists \text{ FIA. (Okos} \sqcap \text{ Szép)}$$

NNF:

$$C_0 = (\leq 1 \text{ FIA}) \sqcap (\exists \text{ FIA. Okos}) \sqcap (\exists \text{ FIA. Szép}) \sqcap \forall \text{ FIA. } (\neg \text{ Okos} \sqcup \neg \text{ Szép})$$

Tabló módszer

b

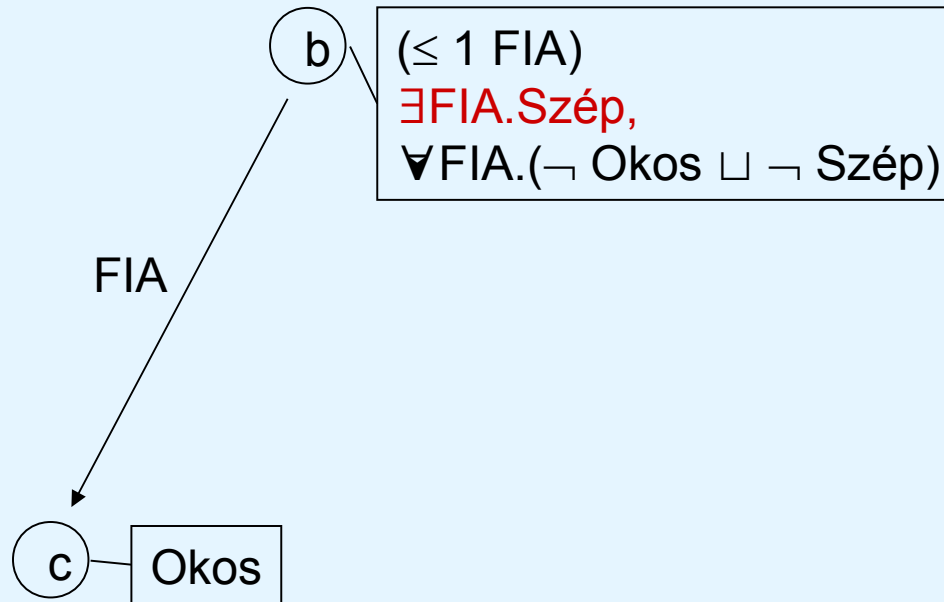
$(\leq 1 \text{ FIA}) \sqcap \exists \text{FIA. Okos} \sqcap \exists \text{FIA. Szép} \sqcap \forall \text{FIA.} (\neg \text{Okos} \sqcup \neg \text{Szép})$

Tabló módszer

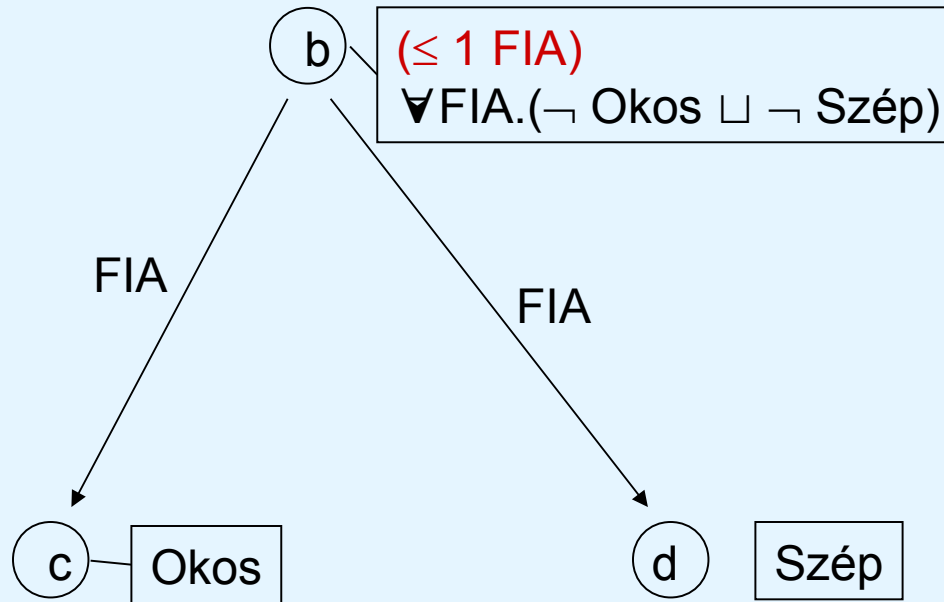
b

$$\begin{aligned} &(\leq 1 \text{ FIA}) \\ &\exists \text{FIA. Okos,} \\ &\exists \text{FIA. Szép,} \\ &\forall \text{FIA. } (\neg \text{Okos} \sqcup \neg \text{Szép}) \end{aligned}$$

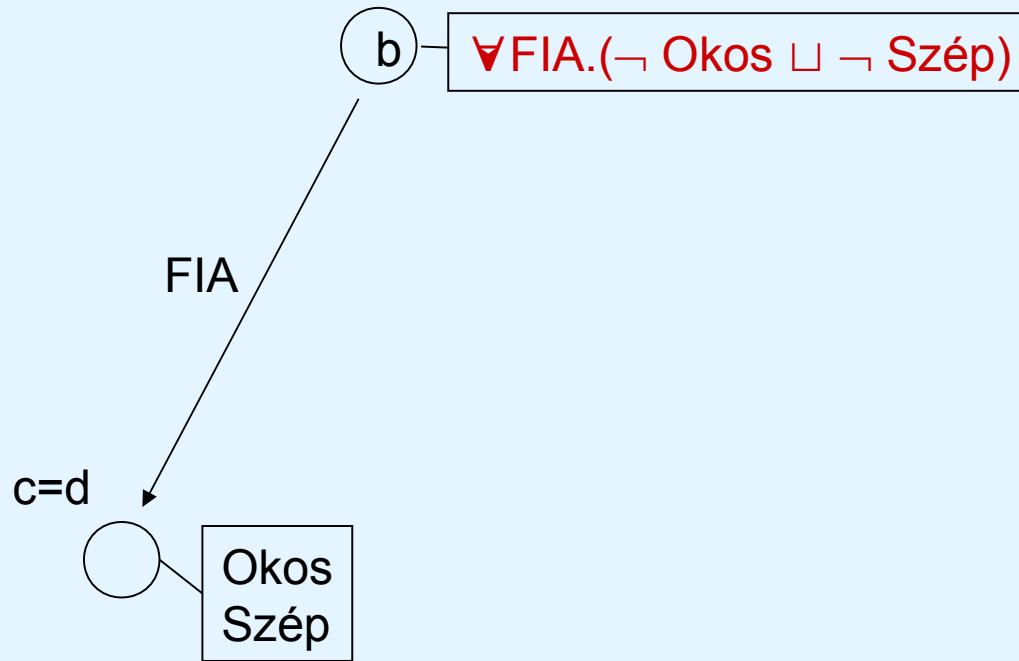
Tabló módszer



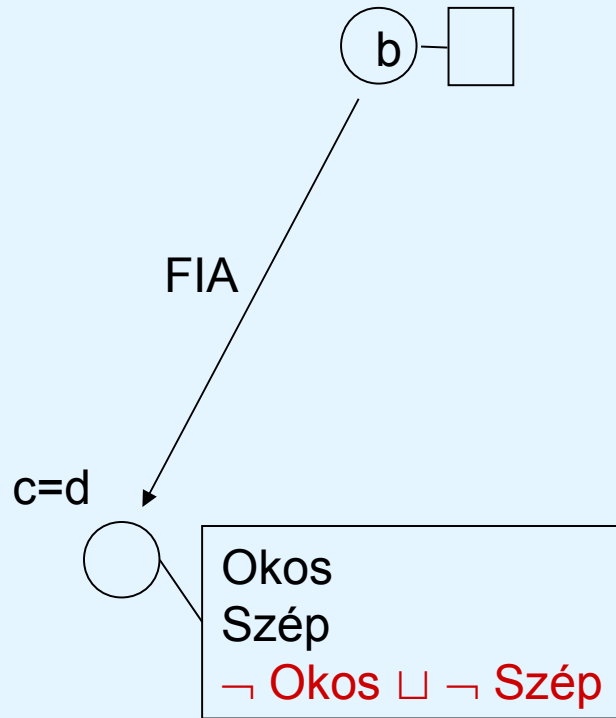
Tabló módszer



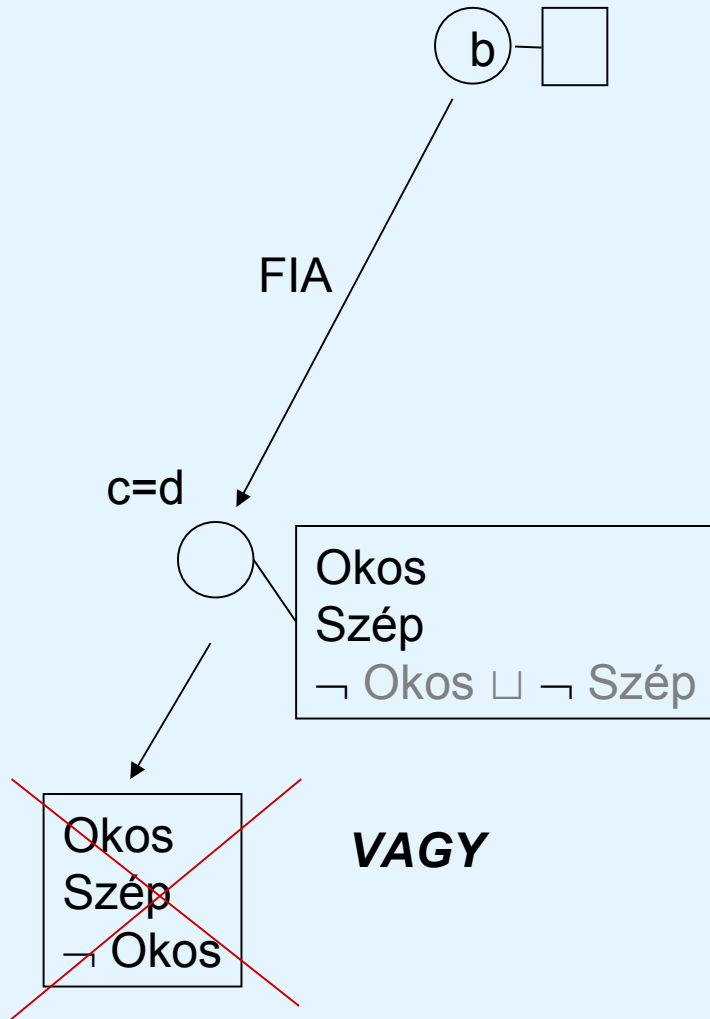
Tabló módszer



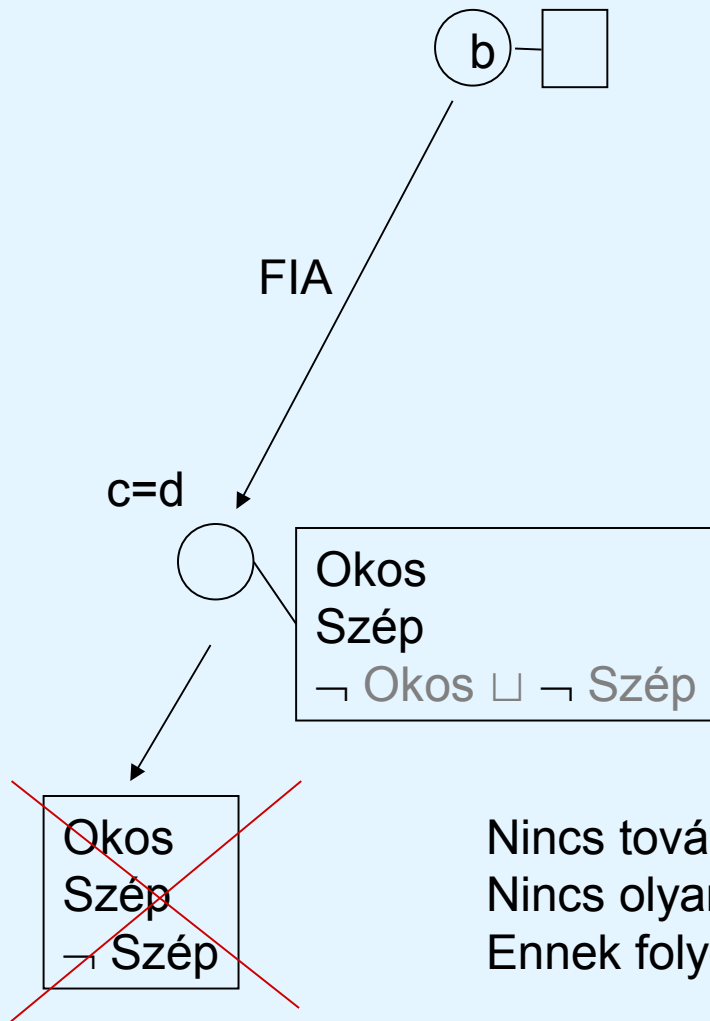
Tabló módszer



Tabló módszer



Tabló módszer



Nincs tovább, minden lehetőségre ellentmondás.
Nincs olyan interpretáció, amely kielégítene az állítást.
Ennek folyamánya, hogy az eredeti tartalmazás igaz.

ALC ... kiegészítő szabályok

$$S \rightarrow_{\sqcap} \{x: C, x: D\} \cup S$$

- ha 1. $x: C \sqcap D$ benne van az S-ben,
2. $x: C$ és $x: D$ nincsenek mind az S-ben.

$$x \bullet \{C_1 \sqcap C_2, \dots\}$$

$$x \bullet \{C_1 \sqcap C_2, \color{red}{C_1}, \color{red}{C_2}, \dots\}$$

$$S \rightarrow_{\sqcup} \{x: E\} \cup S$$

- ha 1. $x: C \sqcup D$ benne van az S-ben,
2. sem $x: C$, sem $x: D$ nincs az S-ben,
3. $E = C$, vagy $E = D$.

$$x \bullet \{C_1 \sqcup C_2, \dots\}$$

$$x \bullet \{C_1 \sqcup C_2, \color{red}{C}, \dots\}$$

for $C \in \{C_1, C_2\}$

$$S \rightarrow_{\exists} \{x R y, y: C\} \cup S$$

- ha 1. $x: \exists R.C$ benne van az S-ben,
2. y egy új változó,
3. nincs olyan z , hogy mind az $x R y$,
mind $z: C$ benne van az S-ben.

$$x \bullet \{\exists R.C, \dots\}$$

$$\begin{array}{c} x \bullet \{\exists R.C, \dots\} \\ \color{red}{R} \downarrow \\ y \bullet \{\color{red}{C}\} \end{array}$$

$$S \rightarrow_{\forall} \{y: C\} \cup S$$

- ha 1. $x: \forall R.C$ benne van az S-ben,
2. $x R y$ benne van az S-ben,
3. $y: C$ nincs benne az S-ben.

$$\begin{array}{c} x \bullet \{\forall R.C, \dots\} \\ \color{blue}{R} \downarrow \\ y \bullet \{\dots\} \end{array}$$

$$\begin{array}{c} x \bullet \{\forall R.C, \dots\} \\ \color{blue}{R} \downarrow \\ y \bullet \{\color{red}{C}, \dots\} \end{array}$$

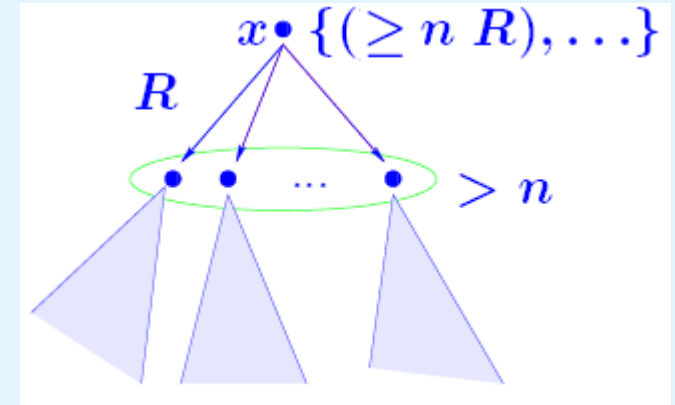
ALC ... kiegészítő szabályok

$x \bullet \{(\geq n R), \dots\}$

x has no R -succ.

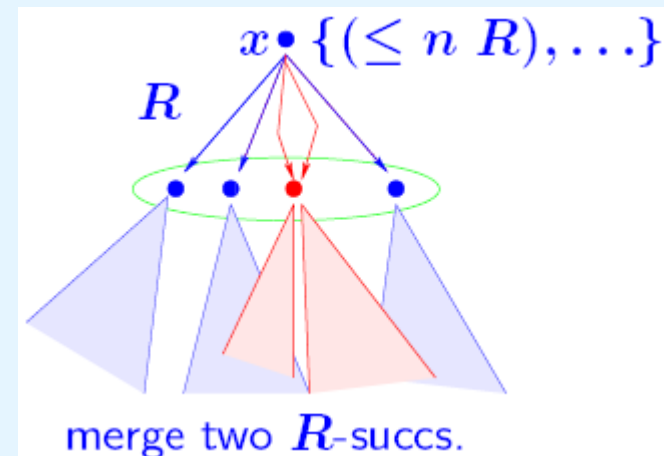
$S \rightarrow_{\geq} \{x R y_k, k=1, \dots, n\} \cup S$

- ha 1. $x: (\geq n R)$ benne van az S -ben,
 2. $x R y_k$ –ből kevesebb van, mint n ,
 akkor n -re bővíteni kell az R utódok
 számát (ha a korlátozás minősített,
 akkor y_k -kat tulajdonsággal kell ellátni.

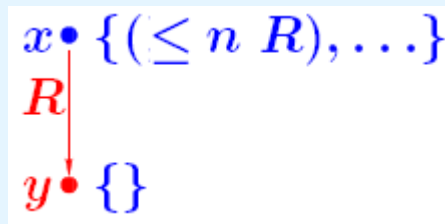


$S \rightarrow_{\leq} \{x R y_k, k=1, \dots, n\} \cup S$

- ha 1. $x: (\leq n R)$ benne van az S -ben,
 2. $x R y_k$ –ből több van, mint n , és vannak
 bennük összevonhatók,
 akkor mindenféle összevonással kísérletezni
 kell, hogy a számuk n alá menjen.



.....



Ütközés

Kényszer rendszer építése közben felismerhetjük az ellentmondásokat, amik a rendszer kielégíthetlenségét jelzik. Ezek az **ütközések**.

Egy ütközés alakja:

$\{x: A, x: \neg A\}$, ahol A egy fogalomnév.

$\{(\geq mR), (\leq nR)\}$, $n < m$

Mivel egy ütközés kielégíthetetlen, az ütközést tartalmazó rendszer is ilyen.

Tabló egyedekkel

Kielégíthető-e az ABox:

$(\text{Szülő} \sqcap \forall \text{GYEREKE.Hímnemű})(\text{jános})$

$\neg \text{Hímnemű}(\text{mária})$

$\text{GYEREKE}(\text{jános}, \text{mária})$

jános: $\text{Szülő} \sqcap \forall \text{GYEREKE.Hímnemű}$

mária: $\neg \text{Hímnemű}$

jános GYEREKE mária

jános : Szülő $\sqcap$ -szabály

jános : $\forall \text{GYEREKE.Hímnemű}$

mária : Hímnemű $\forall$ -szabály

< ÜTKÖZÉS >.

A tudásbázis inkonzisztens

Tabló tulajdonságai (nagyon sok logikára, ACL-től kezdve)

Helyesség (jó megoldást ad, ha azt mondja, hogy ..., akkor ez így van)

A procedúra nem hoz be szükségtelen ellentmondásokat.

Determinisztikus szabályok megtartják a kielégíthetőséget. Nem determinisztikus szabályok esetén mindig van olyan választás, ami a kielégíthetőséget megtartja.

Terminálás (eredményt véges időben ad)

Kényszer rendszer kiegészített (teljes), ha kifogytunk az alkalmazható szabályokból, (az eredeti rendszer kiegészítése) és nincs szó a végtelen szabályláncról.

Teljesség (ha ez így van, akkor az algoritmus is ezt fogja mondani)

Ha S az $\{x: C\}$ kiegészítése és nincs benne ütközés, akkor mindig lehetséges a C egy olyan interpretációjának a megadása, amely mellett a C^I nem üres.

Interpretáció = egy gráf

csomópont = az interpretációs domén egy eleme

csomópont címkéi = az elemet tartalmazó interpretációbeli fogalmak

élek címkéi = relációk elemek között

Nyomok

Kiegészített kényszerrendszer független nyomokra particionálható

– az inkonzisztenciát generáló ütközés csak egy nyomhoz tartozik.

A kiegészített kényszerrendszer egy gráfnak is tekinthető, a nyomok a gyökértől a levelekig vezető pályák.

Ütközés egy levélnél csak a nyomhoz tartozó szabályokból jöhet létre.

Lehetetlen ezt generálni más nyomhoz tartozó szabályokkal

(a kiegészített kényszerrendszer egy fa).

Egy nyom polinomiális méretű!

Funkcionális algoritmusok (heurisztikus keresés)

Kényszerrendszerben csomópontokat csakis a $\rightarrow_{\exists}$ szabály generál.

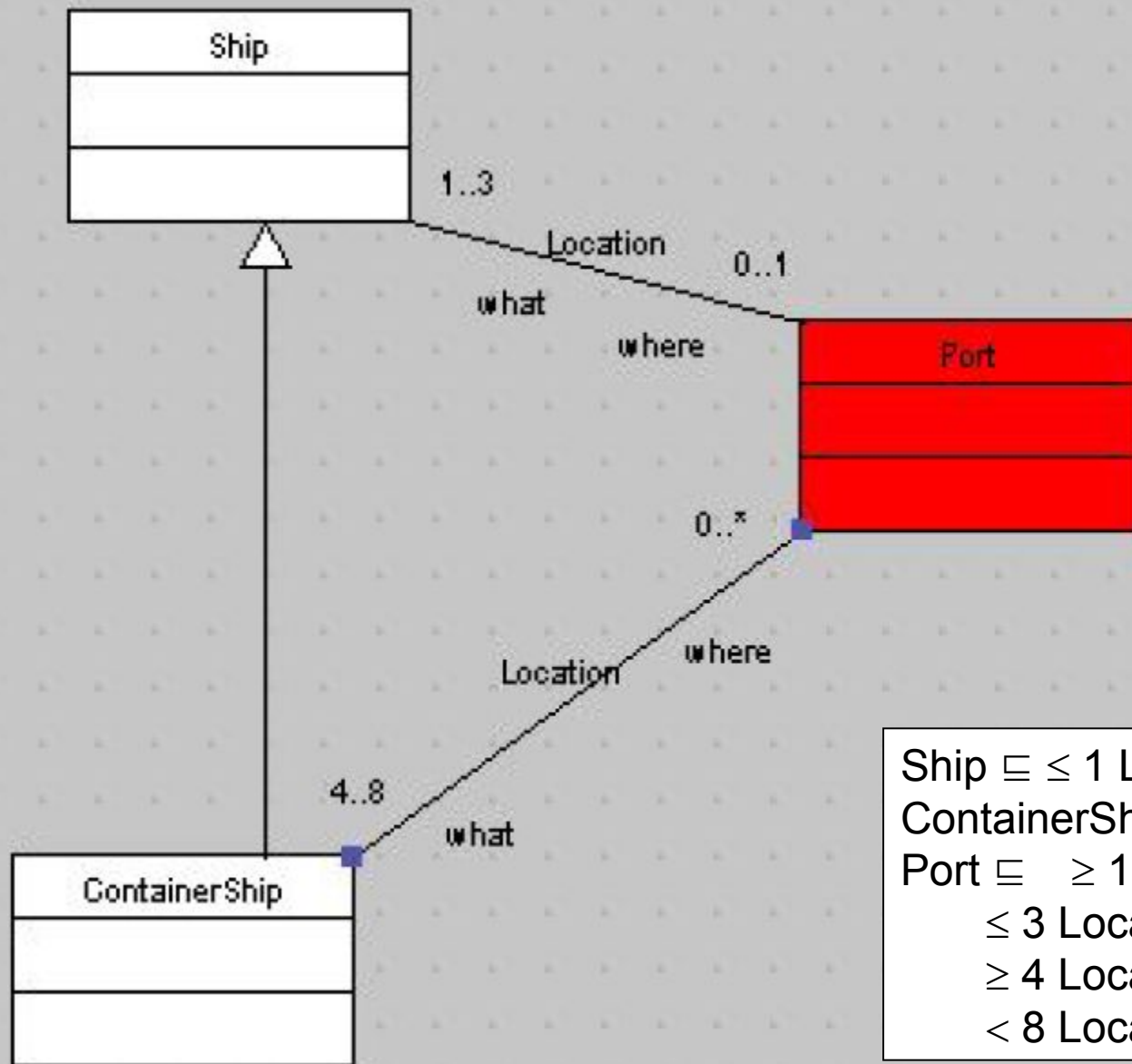
Nyomok kiaknázása érdekében csomópontok generálásánál mélységi kereséssel próbálkozunk.

- a $\rightarrow_{\exists}$ szabályt csak akkor alkalmazzuk, ha nincs más
- ha a $\rightarrow_{\exists}$ szabály több kényszerre alkalmazható, a legújabban generált változóhoz tartozó kényszer a preferált.

Leíró logikák kiterjesztései

- Alapeseti ismeretek és hiedelmek
- Valószínűség és hasonlóság alapú következtetés (Bayes, fuzzy)
- Episztemikus állítások
- Zárt világ feltételezés
- Többeleemes egyedek: rekordok, halmazok, gyűjtemények, aggregációk
- Konkrét domének
- Ontológiai primitív elemek
 - Idő és cselekvés
 - Tér
 - Részek és egész
- Temporális, modális kiterjesztések
- Ontológia editorok, leíró logikai következtetők

...



$\text{Ship} \sqsubseteq \leq 1 \text{ Location.Port}$
 $\text{ContainerShip} \sqsubseteq \text{Ship}$
 $\text{Port} \sqsubseteq \geq 1 \text{ Location}^{-1}.\text{Ship} \sqcap$
 $\leq 3 \text{ Location}^{-1}.\text{Ship} \sqcap$
 $\geq 4 \text{ Location}^{-1}.\text{ContainerShip} \sqcap$
 $< 8 \text{ Location}^{-1}.\text{ContainerShip}$