Logikai programozás

Kovásznai Gergely jegyzete

2007. április 27.

TARTALOMJEGYZÉK

1. ÍTÉLETLOGIKA (NULLADRENDŰ LOGIKA)

1.1. Szintaktika

Jelöljük egy $\mathcal{L}$ ítéletlogikai nyelv ítéletváltozóinak (nem üres) halmazát $\mathcal{V}$-vel.

1.1. Definíció. (Formula)
$\mathcal{L}$ *formulái* induktíve definiáltak a következőképpen:

1. Minden $\mathcal{V}$-beli elem (azaz ítéletváltozó) formula.
2. Ha A és B formulák, akkor

   $\neg A$
   $(A \wedge B)$
   $(A \vee B)$
   $(A \supset B)$
   $(A \equiv B)$

   is formulák.

1.2. Szemantika

Jelöljük az *igaz* és *hamis* igazságértékeket 0-val és 1-gyel.

1.2. Definíció. (Interpretáció)
$\mathcal{L}$ *interpretációja* egy $\mathcal{I} : \mathcal{V} \to \{0, 1\}$ függvény.

A logikai műveletek igazságtáblája:

A	**B**	$\neg A$	$A \wedge B$	$A \vee B$	$A \supset B$	$A \equiv B$
0	**0**	1	0	0	1	1
0	**1**		0	1	1	0
1	**0**	0	0	1	0	0
1	**1**		1	1	1	1

1.3. Definíció. (Formula értéke)
Az $\mathcal{I}$ interpretációban egy F formula *(igazság)értékét* $|F|_{\mathcal{I}}$-vel jelöljük, és a következőképpen definiáljuk:

1. Ha $A \in \mathcal{V}$ (azaz A ítéletváltozó), akkor $|A|_{\mathcal{I}} = \mathcal{I}(A)$.

2. Ha A és B formulák, akkor

$$\begin{aligned}
|\neg A|_{\mathcal{I}} &= \neg|A|_{\mathcal{I}} \\
|A \wedge B|_{\mathcal{I}} &= |A|_{\mathcal{I}} \wedge |B|_{\mathcal{I}} \\
|A \vee B|_{\mathcal{I}} &= |A|_{\mathcal{I}} \vee |B|_{\mathcal{I}} \\
|A \supset B|_{\mathcal{I}} &= |A|_{\mathcal{I}} \supset |B|_{\mathcal{I}} \\
|A \equiv B|_{\mathcal{I}} &= |A|_{\mathcal{I}} \equiv |B|_{\mathcal{I}}
\end{aligned}$$

$\mathcal{I}$ interpretáció és F formula esetén a következő szóhasználattal fogunk élni: ha $|F|_{\mathcal{I}} = 1$, akkor azt mondjuk, hogy „$\mathcal{I}$ kielégíti F-et”. Ennek jelölésére bevezetjük az $\mathcal{I} \models F$ jelölést.

Egy $\mathcal{F}$ *formulahalmaz* esetén akkor használjuk az „$\mathcal{I}$ kielégíti $\mathcal{F}$-et” kifejezést, illetve az $\mathcal{I} \models \mathcal{F}$ jelölést, ha minden $F \in \mathcal{F}$ formulára $\mathcal{I} \models F$.

1.4. Definíció. (Kielégíthetőség/érvényesség/kielégíthetetlenség)

Egy $\mathcal{F}$ formulahalmaz

1. *kielégíthető*, ha $\exists$ olyan $\mathcal{I}$ interpretáció, hogy $\mathcal{I} \models \mathcal{F}$. [1]
2. *logikai törvény* (érvényes), ha $\forall\, \mathcal{I}$ interpretáció esetén $\mathcal{I} \models \mathcal{F}$.
3. *kielégíthetetlen*, ha nem $\exists$ olyan $\mathcal{I}$ interpretáció, hogy $\mathcal{I} \models \mathcal{F}$. [2]

Egy F *formula* kielégíthető/ érvényes/ kielégíthetetlen, ha az $\{F\}$ formulahalmaz kielégíthető/ érvényes/ kielégíthetetlen.

1.5. Definíció. (Logikai következmény)

Egy $\mathcal{P}$ formulahalmaznak *logikai következménye* a K formula, ha $\forall\, \mathcal{I}$ interpretációban teljesül: ha $\mathcal{I} \models \mathcal{P}$, akkor $\mathcal{I} \models K$. Jelölése: $\mathcal{P} \models K$. [3]

1.3. Klózgenerálás

1.6. Definíció. (Literál)

Egy formulát *literálnak* nevezünk, ha A vagy $\neg A$ alakú, ahol A ítéletváltozó. Az A-t a *literál alapjának* nevezzük. Ha a literál A alakú, akkor *pozitív literálról* beszélünk, ha $\neg A$ alakú, akkor pedig *negatív literálról.*

1.7. Definíció. (Klóz)

Klóznak egy $L_1 \vee L_2 \vee \ldots \vee L_k$ formulát nevezünk, ahol $k \geq 0$ és minden L_i literál. Ha $k = 0$, akkor *üres klózról* beszélünk; az üres klózt a $\Box$ jellel jelöljük.

Az $L_1 \vee L_2 \vee \ldots \vee L_k$ klózt akkor elégíti az $\mathcal{I}$ interpretáció, ha *van olyan* L_i, hogy $\mathcal{I} \models L_i$. Ebből következik, hogy $\Box$-t egy interpretáció sem elégíti ki, vagyis $\Box$ *kielégíthetetlen.*

1.8. Definíció. (Klóz normálforma)

Egy formula *klóz normálformában* (másnéven: konjunktív normálforma, KNF) van, ha $C_1 \wedge C_2 \wedge \ldots \wedge C_k$ alakú, ahol $k \geq 0$ és minden C_i klóz.

[1] Azaz van olyan interpretáció, mely $\mathcal{F}$ összes elemét kielégíti.
[2] Azaz $\forall\, \mathcal{I}$ interpretáció esetén $\mathcal{I} \not\models \mathcal{F}$.
[3] $\mathcal{P}$ elemeit premisszáknak, K-t pedig konklúziónak szoktuk nevezni.

Minden formula átírható klóz normálformára a következő szabályok alkalmazásával:

1. Ekvivalenciák és implikációk eliminálása:

$$\begin{aligned} A \equiv B &\sim (A \supset B) \wedge (B \supset A) \\ A \supset B &\sim \neg A \vee B \end{aligned}$$

2. De Morgan azonosságok:

$$\begin{aligned} \neg(A \wedge B) &\sim (\neg A \vee \neg B) \\ \neg(A \vee B) &\sim (\neg A \wedge \neg B) \end{aligned}$$

3. Dupla negációk eliminálása:

$$\neg\neg A \sim A$$

4. Disztributivitás szabályai:

$$\begin{aligned} A \wedge (B \vee C) &\sim (A \wedge B) \vee (A \wedge C) \\ A \vee (B \wedge C) &\sim (A \vee B) \wedge (A \vee C) \end{aligned}$$

Egy KNF-ben levő $C_1 \wedge C_2 \wedge \ldots \wedge C_k$ formula kielégíthetőségét (kielégíthetetlenségét) vizsgáljuk. Ennek eldöntése megegyezik a $\{C_1, C_2, \ldots, C_k\}$ klózhalmaz kielégíthetőségének (kielégíthetetlenségének) vizsgálatával.

A fentiek értelmében tehát bármely F formula átalakítható egy $\mathcal{C}(F)$ klózhalmazzá, amelyre a következő igaz:

F akkor és csak akkor kielégíthető (kielégíthetetlen),
ha $\mathcal{C}(F)$ is kielégíthető (kielégíthetetlen).

1.9. Példa.

Igazoljuk a következő következtetés helyességét:

Ha vonattal megyek, akkor ha a vonat késik, lekésem a találkozót.

Ha lekésem a találkozót, és rossz kedvem lesz, akkor nem megyek holnap kirándulni.

Ha nem kapom meg az állást, akkor rossz kedvem lesz, és elmegyek holnap kirándulni.

Tehát ha vonattal megyek, akkor ha a vonat késik, megkapom az állást.

Vezessük be a következő ítéletváltozókat, azaz legyen $\mathcal{V}$ a következő:

$$\mathcal{V} = \{A, H, K, M, R, T\}$$

Ezen ítéletváltozók informális jelentése a következő:

A: „megkapom az állást”

H: „holnap elmegyek kirándulni”

K: „a vonat késik”

M: „vonattal megyek”

R: „rossz kedvem lesz”

T: „lekésem a találkozót”

Írjuk fel a mondatokat formulákként:

1. „Ha vonattal megyek, akkor ha a vonat késik, lekésem a találkozót.” $\longrightarrow M \wedge K \supset T$
2. „Ha lekésem a találkozót, és rossz kedvem lesz, akkor nem megyek holnap kirándulni.” $\longrightarrow T \wedge R \supset \neg H$
3. „Ha nem kapom meg az állást, akkor rossz kedvem lesz, és elmegyek holnap kirándulni.” $\longrightarrow \neg A \supset R \wedge H$
4. „Ha vonattal megyek, akkor ha a vonat késik, megkapom az állást.” $\longrightarrow M \wedge K \supset A$

Alakítsuk ezeket a formulákat KNF-re:

1. $M \wedge K \supset T \;\longrightarrow\; \neg(M \wedge K) \vee T \;\longrightarrow\; \neg M \vee \neg K \vee T$
2. $T \wedge R \supset \neg H \;\longrightarrow\; \neg(T \wedge R) \vee \neg H \;\longrightarrow\; \neg T \vee \neg R \vee \neg H$
3. $\neg A \supset R \wedge H \;\longrightarrow\; A \vee (R \wedge H) \;\longrightarrow\; (A \vee R) \wedge (A \vee H)$
4. A konklúziónak, azaz az $M \wedge K \supset A$ formulának a negáltját kell KNF-re átalakítani (lásd: 1.10. lemma).

   Tehát: $\neg(M \wedge K \supset A) \;\longrightarrow\; \neg(\neg M \vee \neg K \vee A) \;\longrightarrow\; M \wedge K \wedge \neg A$

A keresett $\mathcal{S}$ klózhalmaz a kapott KNF-ben levő formulák klózaiból áll, azaz:

$$\mathcal{S} \;=\; \left\{ \begin{array}{c} \neg M \vee \neg K \vee T \\ \neg T \vee \neg R \vee \neg H \\ A \vee R \\ A \vee H \\ M \\ K \\ \neg A \end{array} \right\}$$

1.4. Rezolúció

A rezolúció egy ún. *cáfoló* tételbizonyító módszer. Ez azt jelenti, hogy ha adottak a $P_1, \ldots, P_k$ premisszák és a K konklúzió, akkor $\{P_1, \ldots, P_k\} \models K$ igazolásához azt próbáljuk belátni, hogy a $\{P_1, \ldots, P_k, \neg K\}$ formulahalmaz *kielégíthetetlen.*

1.10. Lemma.
$\{P_1, \ldots, P_k\} \models K$ *akkor és csak akkor, ha* $\{P_1, \ldots, P_k, \neg K\}$ *kielégíthetetlen.*

Bizonyítás.
$\{P_1, \ldots, P_k\} \models K$ akkor és csak akkor, ha a $P_1 \wedge \ldots \wedge P_k \supset K$ formula logikai törvény. Ez utóbbival ekvivalens annak belátása, hogy $\neg(P_1 \wedge \ldots \wedge P_k \supset K)$ kielégíthetetlen. Az utóbbi formula átalakítható (a de Morgan azonosságok alkalmazásával) a $P_1 \wedge \ldots P_k \wedge \neg K$ alakra; tehát ezen formuláról kell belátnunk, hogy kielégíthetetlen. Ezzel ekvivalens annak igazolása, hogy a $\{P_1, \ldots, P_k, \neg K\}$ formulahalmaz kielégíthetetlen.

Mivel minden F formula átalakítható egy $\mathcal{C}(F)$ klózhalmazzá, így igaz a következő:

a $\{P_1, \ldots, P_k, \neg K\}$ formulahalmaz akkor és csak akkor kielégíthetetlen,
ha $\{\mathcal{C}(P_1), \ldots, \mathcal{C}(P_k), \mathcal{C}(\neg K)\}$ is kielégíthetetlen.

Így vezettük vissza a logikai következmény kérdésének eldöntését egy klózhalmaz kielégíthetetlenségének vizsgálatára.

1.11. Definíció. (Rezolvens)

Tekintsük a $C_1 = L_1 \vee C_1'$ és $C_2 = L_2 \vee C_2'$ klózokat, ahol L_1 és L_2 azonos alapú és ellentétesen negált literálok. A $C_1' \vee C_2'$ klózt a C_1 és C_2 *rezolvensének* nevezzük.

1.12. Lemma.

Ha C' a C_1 és C_2 klózok rezolvense, akkor $\{C_1, C_2\} \models C'$.

1.13. Definíció. (Rezolúciós levezetés)

Egy $\mathcal{S}$ klózhalmazból való *rezolúciós levezetés* alatt klózoknak egy véges $C_1, C_2, \ldots, C_n$ $(n \geq 1)$ sorozatát értjük, ahol minden C_i esetén

1. vagy $C_i \in \mathcal{S}$,
2. vagy C_i rezolvense a C_{j_1} és C_{j_2} klózoknak, ahol $j_1, j_2 < i$.

Rezolúciós cáfolatnak nevezzük a fenti rezolúciós levezetést, ha $C_n = \Box$.

A rezolúciós kalkulus eldöntésproblémája az, hogy egy $\mathcal{S}$ klózhalmazhoz konstruálható-e rezolúciós cáfolat; azaz hogy $\mathcal{S}$-ből *levezethető-e az üres klóz.* A rezolúció *helyes és teljes kalkulus.* Ezen fogalmak jelentése:

- helyesség: ha $\mathcal{S}$-hez van rezolúciós cáfolat, akkor $\mathcal{S}$ kielégíthetetlen.
- teljesség: ha $\mathcal{S}$ kielégíthetetlen, akkor van $\mathcal{S}$-hez rezolúciós cáfolat.

Megadjuk a helyesség bizonyítását:

1.14. Tétel. (Rezolúció helyessége)

A rezolúciós kalkulus helyes, azaz ha az $\mathcal{S}$ klózhalmaznak van rezolúciós cáfolata, akkor $\mathcal{S}$ kielégíthetetlen.

Bizonyítás.

Legyen az $\mathcal{S}$-hez konstruált rezolúciós cáfolat $C_1, C_2, \ldots, C_n$.

Először be fogjuk látni, hogy minden C_i esetén $\mathcal{S} \models C_i$ (azaz a rezolúciós cáfolat minden eleme logikai következménye $\mathcal{S}$-nek). Vagyis belátjuk, hogy minden olyan $\mathcal{I}$ interpretáció esetén, melyre $\mathcal{I} \models \mathcal{S}$ teljesül, az is teljesül, hogy $\mathcal{I} \models C_i$ minden C_i-re. Ennek bizonyítása induktív:

1. Ha $C_i \in \mathcal{S}$, akkor ez triviálisan teljesül.
2. Ha C_i a C_{j_1} és C_{j_2} rezolvense, akkor:
   (a) Az induktív feltevésünk az, hogy az állítás teljesül C_{j_1}-re és C_{j_2}-re is. Azaz $\mathcal{S} \models C_{j_1}$ és $\mathcal{S} \models C_{j_2}$.
   (b) Az 1.12. lemma alapján tudjuk, hogy $\{C_{j_1}, C_{j_2}\} \models C_i$, ami miatt pedig (az induktív feltevést felhasználva) teljesül az $\mathcal{S} \models C_i$ összefüggés.

Ezek után már könnyű belátni, hogy $\mathcal{S}$ kielégíthetetlen. Mivel $C_n = \Box$, így a fentiek miatt $\mathcal{S} \models \Box$. Azaz nem létezhet olyan interpretáció, mely kielégíti $\mathcal{S}$-t (mivel $\Box$-t semelyik interpretáció sem elégíti ki). Vagyis $\mathcal{S}$ kielégíthetetlen.

A teljesség bizonyítása bonyolultabb, ez nem tartozik a tematikába.

1.5. Lineáris inputrezolúció

1.15. Definíció. (Lineáris rezolúciós levezetés)

Egy *lineáris rezolúciós levezetés* alatt olyan $C_1, M_1, C_2, M_2, \ldots, C_{n-1}, M_{n-1}, C_n$ rezolúciós levezetést értünk, melyben minden $i = 2, 3, \ldots, n$ esetén C_i a C_{i-1} és M_{i-1} klózok rezolvense.

A fenti C_i klózokat *centrális klózoknak*, az M_i-ket *mellékklózoknak* nevezzük. A lineáris rezolúciós kalkulus is *helyes és teljes*.

A lineáris rezolúcióban a centrális klózokra tettünk megszorítást: ezeknek a sorrendben megelőző két klóz rezolvensének kell lenniük. Felmerül az igény, a mellékklózokra is tegyünk valamilyen megszorítást.

1.16. Definíció. (Lineáris inputrezolúciós levezetés)

Egy *lineáris inputrezolúciós levezetés* alatt olyan $C_1, M_1, C_2, M_2, \ldots, C_{n-1}, M_{n-1}, C_n$ lineáris rezolúciós levezetést értünk, melyben minden $M_i \in \mathcal{S}$.

A lineáris inputrezolúciós kalkulus *helyes*. Viszont belátható, hogy *nem teljes*, egy egyszerű ellenpéldán keresztül.

1.17. Példa.

Tekintsük a következő klózhalmazt:

$$\mathcal{S} = \left\{ \begin{array}{c} A \vee B \ , \\ \neg A \vee B \ , \\ \neg A \vee \neg B \ , \\ A \vee \neg B \end{array} \right\}$$

Könnyű belátni, hogy $\mathcal{S}$ kielégíthetetlen (pl. igazságtáblával). Mégsem tudjuk belőle levezetni lineáris inputrezolúcióval az üres klózt. Ennek az az oka, hogy nincs egy olyan $\mathcal{S}$-beli klóz (másnéven: input klóz) sem, mely csupán egyetlen literált tartalmazna. Így – mivel a mellékklózok mindig input klózok kell legyenek – a levezetés során egyszer sem fordulhat elő, hogy valamelyik előállított rezolvens az üres klóz lesz.

Belátható viszont, hogy a lineáris inputrezolúció *teljes a Horn-logikában*, azaz ún. Horn-klózok halmazain.

1.18. Definíció. (Horn-klóz)

A *Horn-klóz* olyan klóz, mely legfeljebb 1 db. pozitív literált tartalmaz.

1.19. Példa.

Az 1.9. példában előállított

$$\mathcal{S} = \left\{ \begin{array}{c} \neg M \vee \neg K \vee T \\ \neg T \vee \neg R \vee \neg H \\ A \vee R \\ A \vee H \\ M \\ K \\ \neg A \end{array} \right\}$$

klózhalmazról fogjuk most megmutatni, hogy kielégíthetetlen. Ezt *lineáris inputrezolúcióval* próbáljuk bizonyítani. Ám mivel $\mathcal{S}$-ben *nem csak Horn-klózok* vannak ($A \vee R$ és $A \vee H$ nem ilyenek), így még ha $\mathcal{S}$

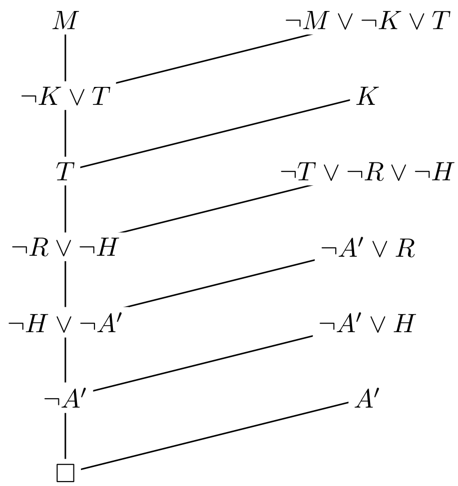
kielégíthetetlen is lenne, sem biztos, hogy le tudjuk belőle vezetni lineáris inputrezolúcióval az üres klózt. Ezért megpróbálhatjuk átalakítani $\mathcal{S}$-t oly módon, hogy csak Horn-klózokat tartalmazzon. Ez könnyen megtehető egy új ítéletváltozó bevezetésével:

A': „nem kapom meg az állást”

Azaz $A' \sim \neg A$. Így átírható $\mathcal{S}$ a következőképpen:

$$\mathcal{S} \quad = \quad \left\{ \begin{array}{c} \neg M \vee \neg K \vee T \\ \neg T \vee \neg R \vee \neg H \\ \neg A' \vee R \\ \neg A' \vee H \\ M \\ K \\ A' \end{array} \right\}$$

Látható, hogy $\mathcal{S}$ így már csak Horn-klózokat tartalmaz. Adjunk meg tehát $\mathcal{S}$-hez lineáris inputrezolúciós cáfolatot, a következő formában:



A levezetés fenti ábrázolásában a fa bal oldali csúcsai a centrális klózok, jobb oldali csúcsai a mellékklózok. Az egy szinten levő két klózt rezolváljuk egymással, és kapjuk a velük éllel összekötött centrális klózt.

1.20. Megjegyzés.

A fenti példában szereplő lineáris inputrezolúciós levezetés alulról harmadik sorában a $\neg H \vee \neg A'$ és az $\neg A' \vee H$ klózok rezolválásakor a kapott rezolvens elvileg $\neg A' \vee \neg A'$; azonban a klózokban *az azonos literálokat összevonjuk.*

2. PREDIKÁTUMLOGIKA (ELSŐRENDŰ LOGIKA)

2.1. Szintaktika

Egy $\mathcal{L}$ predikátumlogikai nyelv megadható a következő halmazok segítségével:

1. *Változók* megszámlálhatóan végtelen $\mathcal{V}$ halmaza.
2. *Függvényszimbólumok* (esetleg üres) $\mathcal{F}n$ halmaza.
3. *Predikátumszimbólumok* nem üres $\mathcal{P}r$ halmaza.

Minden függvény- és predikátumszimbólumhoz megadunk továbbá egy-egy $k \geq 0$ egész számot, melyet az adott szimbólum *fokának* nevezünk. [1]

Az adott nyelven szintaktikailag helyes *kifejezéseket* tudunk felírni.

2.1. Definíció. (Kifejezés – term és formula)

$\mathcal{L}$ *kifejezései* két csoportra oszthatók: termekre és formulákra. Ezek a következőképpen, induktíve definiáltak:

1. *Term*:
   (a) Minden $\mathcal{V}$-beli elem (azaz változó) term.
   (b) Minden $f(t_1, \ldots, t_k)$ alakú kifejezés term, ahol $f \in \mathcal{F}n$ k-fokú függvényszimbólum és $t_1, \ldots, t_k$ termek.
2. *Formula*:
   (a) Minden $P(t_1, \ldots, t_k)$ alakú kifejezés formula, ahol $P \in \mathcal{P}r$ k-fokú predikátumszimbólum és $t_1, \ldots, t_k$ termek. Az ilyen alakú formulákat *atomi formuláknak* nevezzük.
   (b) Ha A és B formulák, akkor

   $\neg A$
   $(A \wedge B)$
   $(A \vee B)$
   $(A \supset B)$
   $(A \equiv B)$

   is formulák.
   (c) Ha A formula és $x \in \mathcal{V}$, akkor

   $\forall x A$
   $\exists x A$

   is formulák.

[1] Mint látható, nem definiálunk a nyelvben *típusok halmazát*; tulajdonképpen feltesszük, hogy csak egyfajta típus van a nyelvben (ez nem jelent lényeges megszorítást). Hasonlóképpen, nem definiálunk *konstansszimbólumokat* a nyelvben; ezeknek tulajdonképpen a 0-fokú függvényszimbólumok felelnek meg.

Egy kifejezésben egy változó-előfordulás *kötött*, ha egy őt kötő kvantor hatáskörébe esik; egyébként *szabad*.

Egy F kifejezés *szabad változóinak halmazát* $\mathcal{FV}(F)$-fel jelöljük. Egy F kifejezés *zárt*, ha $\mathcal{FV}(F) = \emptyset$; egyébként *nyílt*.

A következőkben gyakran fogunk használni változókon értelmezett függvényeket; azaz olyan σ függvényeket, ahol $\mathcal{D}om(\sigma) \subseteq \mathcal{V}$. [2] Az ilyen függvényeket sokszor

$$\sigma \quad = \quad \begin{bmatrix} x_1 & x_2 & \dots & x_k \\ \sigma(x_1) & \sigma(x_2) & \dots & \sigma(x_k) \end{bmatrix}$$

táblázatos alakban fogjuk ábrázolni, ahol $\mathcal{D}om(\sigma) = \{x_1, \dots, x_k\}$. Ha $\mathcal{D}om(\sigma) = \emptyset$, akkor egy ilyen (üres) függvény jelölésére az ϵ-t használjuk.

2.2. Definíció. (Helyettesítés)

(Term)helyettesítés alatt egy változókon értelmezett függvényt értünk, melynek az értelmezési tartománya az (összes) termek halmaza.

Egy F kifejezés és egy σ helyettesítés esetén $F\sigma$ alatt azt a kifejezést értjük, melyet úgy kapunk, hogy F-ben minden $x \in \mathcal{D}om(\sigma)$ változó minden *szabad* előfordulását (egyidejűleg) felülírjuk $\sigma(x)$-szel.

2.2. Szemantika

2.3. Definíció. (Interpretáció)

Az $\mathcal{L}$ nyelv $\mathcal{I}$ *interpretációja* egy nem üres $\mathcal{U}$ halmaz (ún. *univerzum*) és a következő függvények segítségével adható meg:

1. Minden $f \in \mathcal{F}n$ k-fokú függvényszimbólumhoz rendelünk egy

   $$f^{\mathcal{I}} : \underbrace{U \times U \times \cdots \times U}_{k} \mapsto U$$

   függvényt.

2. Minden $P \in \mathcal{P}r$ k-fokú predikátumszimbólumhoz rendelünk egy

   $$P^{\mathcal{I}} : \underbrace{U \times U \times \cdots \times U}_{k} \mapsto \{0,1\}$$

   (logikai) függvényt.

2.4. Definíció. (Változó-kiértékelés)

A fenti $\mathcal{I}$ interpretációban egy *változó-kiértékelés* alatt egy olyan változókon értelmezett függvényt értünk, melynek értékkészlete $\mathcal{U}$.

Egy F *kifejezés változó-kiértékelése* olyan κ változó-kiértékelés, melynek értelmezési tartománya $\mathcal{FV}(F)$.

Valamely x változó és valamely $u \in \mathcal{U}$ esetén $\kappa\begin{bmatrix} x \\ u \end{bmatrix}$ alatt olyan κ' változó-kiértékelést értünk, hogy

- $Dom(\kappa') = Dom(\kappa) \cup \{x\}$,
- $\kappa'(x) = u$ és
- $\kappa'(y) = \kappa(y)$ minden $y \neq x$ esetén.

[2] $\mathcal{D}om(\sigma)$-val jelöljük a σ értelmezési tartományát.

2.5. Definíció. (Kifejezés értéke változó-kiértékelés mellett)
Legyen F egy kifejezés. Legyen $\mathcal{I}$ egy interpretáció, és legyen κ az F-nek egy változó-kiértékelése.
$\mathcal{I}$-ben κ *mellett az* F *értékét* $|F|^{\kappa}_{\mathcal{I}}$-val jelöljük, és a következőképpen definiáljuk:

1. Term esetén:

   (a) Ha $x \in \mathcal{V}$ változó, akkor

   $$|x|^{\kappa}_{\mathcal{I}} = \kappa(x)$$

   (b) Ha $f \in \mathcal{F}n$ függvényszimbólum, akkor

   $$|f(t_1, \ldots, t_k)|^{\kappa}_{\mathcal{I}} = f^{\mathcal{I}}(|t_1|^{\kappa}_{\mathcal{I}}, \ldots, |t_k|^{\kappa}_{\mathcal{I}})$$

2. Formula esetén:

   (a) Ha $P \in \mathcal{P}r$ predikátumszimbólum, akkor

   $$|P(t_1, \ldots, t_k)|^{\kappa}_{\mathcal{I}} = P^{\mathcal{I}}(|t_1|^{\kappa}_{\mathcal{I}}, \ldots, |t_k|^{\kappa}_{\mathcal{I}})$$

   (b) Ha A és B formulák, akkor

   $$\begin{aligned} |\neg A|^{\kappa}_{\mathcal{I}} &= \neg |A|^{\kappa}_{\mathcal{I}} \\ |A \wedge B|^{\kappa}_{\mathcal{I}} &= |A|^{\kappa}_{\mathcal{I}} \wedge |B|^{\kappa}_{\mathcal{I}} \\ |A \vee B|^{\kappa}_{\mathcal{I}} &= |A|^{\kappa}_{\mathcal{I}} \vee |B|^{\kappa}_{\mathcal{I}} \\ |A \supset B|^{\kappa}_{\mathcal{I}} &= |A|^{\kappa}_{\mathcal{I}} \supset |B|^{\kappa}_{\mathcal{I}} \\ |A \equiv B|^{\kappa}_{\mathcal{I}} &= |A|^{\kappa}_{\mathcal{I}} \equiv |B|^{\kappa}_{\mathcal{I}} \end{aligned}$$

   (c) Ha A formula és $x \in \mathcal{V}$ változó, akkor

   $$|\forall x A|^{\kappa}_{\mathcal{I}} = 1 \quad \Leftrightarrow \quad \text{minden } u \in \mathcal{U} \text{ esetén } |A|_{\mathcal{I}}^{\kappa\left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right]} = 1$$

   $$|\exists x A|^{\kappa}_{\mathcal{I}} = 1 \quad \Leftrightarrow \quad \text{létezik olyan } u \in \mathcal{U}\text{, hogy } |A|_{\mathcal{I}}^{\kappa\left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right]} = 1$$

2.6. Definíció. (Formula értéke)
Az $\mathcal{I}$ interpretációban az F formula *értékét* $|F|_{\mathcal{I}}$-vel jelöljük, és a következőképpen definiáljuk:

$$|F|_{\mathcal{I}} = 1 \quad \Leftrightarrow \quad |\forall x_1 \ldots \forall x_k F|^{\epsilon}_{\mathcal{I}}$$

ahol $\mathcal{FV}(F) = \{x_1, \ldots, x_k\}$.

Azaz egy F formula akkor és csak akkor igaz egy interpretációban, ha F minden változókiértékelése mellett igaz F az interpretációban.

A kielégíthetőség/ érvényesség/ kielégíthetetlenség és logikai következmény fogalma és definíciója azonos az 1.2. fejezetben írtakkal.

2.3. Klózgenerálás

2.7. Definíció. (Literál)
Egy formulát *literálnak* nevezünk, ha A vagy $\neg A$ alakú, ahol A *atomi formula*[3]. Az A-t a *literál alapjának* nevezzük. Ha a literál A alakú, akkor *pozitív literálról* beszélünk, ha $\neg A$ alakú, akkor pedig *negatív literálról.*

[3] Csak ebben az egy szóban tér el a nulladrendű literál 1.6. definíciójától.

A klóz és a klóz normálforma definíciója megegyezik az 1.7., illetve az 1.8. definíciókkal.

Hasonlóképpen, mint ítéletlogikában, minden formula átírható klóz normálformára. Ezen átalakításhoz ugyanazokat a szabályokat alkalmazzuk, mint az 1.3. fejezetben, továbbá még a következő szabályokat:

1. Kvantoros de Morgan azonosságok:

$$\begin{aligned} \neg\forall xA &\sim \exists x\neg A \\ \neg\exists xA &\sim \forall x\neg A \end{aligned}$$

2. (Egyoldali) kvantorkiemelés szabályai:

$$\begin{aligned} (\bigcirc xA \circ B) &\sim \bigcirc x'(A \circ B) \\ (A \circ \bigcirc xB) &\sim \bigcirc x'(A \circ B) \end{aligned}$$

   ahol $\bigcirc \in \{\forall, \exists\}$, $\circ \in \{\wedge, \vee\}$ és x' új változó[4].

Ezen szabályokat alkalmazva tetszőleges formulát ekvivalens $\bigcirc x_1 \cdots \bigcirc x_k A$ alakra tudunk hozni, ahol A KNF-ben van. Ezen a ponton azonban nem állunk meg. A prenex alakban szereplő összes egzisztenciális kvantort elimináljuk ún. Skolem-függvények bevezetésével:

2.8. Definíció. (Skolemizált)

Egy $\forall x_1 \ldots \forall x_k \exists y A$ formula[5] ($k \geq 0$) skolemizáltja az $\forall x_1 \ldots \forall x_k A\left[\frac{y}{f(x_1,\ldots,x_k)}\right]$ formula, ahol f új függvényszimbólum[6].

2.9. Lemma.

Egy formula skolemizáltja akkor és csak akkor kielégíthető (kielégíthetetlen), ha maga a formula is kielégíthető (kielégíthetetlen).

A fentiek alapján bármely formula kielégíthetőségét (kielégíthetetlenségét) vizsgálhatjuk úgy, hogy a formulát először $\forall x_1 \ldots \forall x_m (C_1 \wedge \ldots \wedge C_k)$ alakra hozzuk – ahol minden C_i klóz –, majd ezen formulának vizsgáljuk a kielégíthetőségét (kielégíthetetlenségét). Ez definíció szerint megegyezik a $C_1 \wedge \ldots \wedge C_k$ formula, azaz a $\{C_1, C_2, \ldots, C_k\}$ klózhalmaz kielégíthetőségének (kielégíthetetlenségének) vizsgálatával.

2.10. Példa.

Igazoljuk a következő következtetés helyességét:

> Néhány kitüntetett kedvence az elnöknek vagy az igazgatónak.
>
> Az igazgató kedvencei közül csak olyanokat tüntettek ki, akik az elnöknek is kedvencei.
>
> Tehát az elnök kedvencei között akad kitüntetett.

Bevezetjük a következő predikátumszimbólumokat, azaz legyen a $\mathcal{P}r$ halmaz a következő:

$$\mathcal{P}r = \{E^{(1)}, I^{(1)}, K^{(1)}\} \quad ,$$

ahol a predikátumszimbólumok felsőindexébe írtuk az adott predikátumszimbólum fokát. Ezen predikátumszimbólumok informális jelentése a következő:

---

[4] Azaz a változók $\mathcal{V}$ halmazát kiegészítjük x'-vel. Ha mindig új változót vezetünk be, akkor ezzel kiküszöbölhető a változótiszta alakra hozás a klózgenerálás lépései közül.

[5] Itt A tetszőleges formula. Azaz akár kvantorokat is tartalmazhat.

[6] Azaz a függvényszimbólumok $\mathcal{F}n$ halmazát kiegészítjük f-fel.

$E(x)$: „x kedvence az elnöknek”

$I(x)$: „x kedvence az igazgatónak”

$K(x)$: „x-et kitüntették”

Függvényszimbólumokra nem lesz szükségünk, azaz $\mathcal{F}n = \emptyset$. Írjuk fel a mondatokat formulákként:

1. „Néhány kitüntetett kedvence az elnöknek vagy az igazgatónak.” $\longrightarrow \exists x\big(K(x) \wedge (E(x) \vee I(x))\big)$
2. „Az igazgató kedvencei közül csak olyanokat tüntettek ki, akik az elnöknek is kedvencei.” $\longrightarrow$ $\forall y\big(I(y) \wedge K(y) \supset E(y)\big)$
3. „Az elnök kedvencei között akad kitüntetett.” $\longrightarrow \exists z(E(z) \wedge K(z))$

Alakítsuk ezeket a formulákat (prenex, skolemizált) KNF-re:

1. $\exists x\big(K(x) \wedge (E(x) \vee I(x))\big) \;\longrightarrow\; K(c) \wedge (E(c) \vee I(c))$
2. $\forall y\big(I(y) \wedge K(y) \supset E(y)\big) \;\longrightarrow\; \forall y\big(\neg I(y) \vee \neg K(y) \vee E(y)\big)$
3. A konklúziónak, azaz a $\exists z(E(z) \wedge K(z))$ formulának a negáltját kell KNF-re átalakítani (lásd: 1.10. lemma).

   Tehát: $\neg\exists z(E(z) \wedge K(z)) \;\longrightarrow\; \forall z\big(\neg E(z) \vee \neg K(z)\big)$

A fenti 1. pontban skolemizáltunk, azaz az x változó helyére a c új konstansszimbólumot (0-fokú függvényszimbólumot) helyettesítettünk.

A keresett $\mathcal{S}$ klózhalmaz a kapott KNF-ben levő formulák klózaiból áll, azaz:

$$\mathcal{S} \;=\; \left\{ \begin{array}{c} K(c) \\ E(c) \vee I(c) \\ \neg I(y) \vee \neg K(y) \vee E(y) \\ \neg E(z) \vee \neg K(z) \end{array} \right\}$$

2.4. Rezolúció

Elsőrendű logikában a rezolúció kevés ponton módosul az 1.4. fejezetben írtakhoz képest. Először is ismertetjük az unifikátor[7] fogalmát.

2.11. Definíció. (Unifikátor)
Egy A és egy B formula *unifikátora* egy olyan σ helyettesítés, melyre $A\sigma = B\sigma$.

Két azonos alapú, pozitív L_1 és L_2 literál *legáltalánosabb* unifikátorának kiszámítására Robinson adott algoritmust:

1. $\sigma := \epsilon$
2. Ha nincs eltérő karakter L_1-ben és L_2-ben, akkor a keresett unifikátor σ, és vége.
3. Jelölje t_1, illetve t_2 az első eltérő karakteren kezdődő termet az L_1-ben, illetve az L_2-ben.
4. Ha t_1 és t_2 egyike sem változó (azaz függvényszimbólummal kezdődnek), akkor nincs unifikátor, és vége.
5. Ha t_1 változó, akkor vezessük be a következő jelöléseket: $x := t_1$ és $t := t_2$. Egyébként legyen $x := t_2$ és $t := t_1$.

[7] másnéven: illesztő helyettesítés

6. *Occur Check:* ha $x \in \mathcal{FV}(t)$, akkor nincs unifikátor, és vége.
7. $\sigma := \sigma \begin{bmatrix} x \\ t \end{bmatrix}$
8. $L_1 := L_1\sigma$ és $L_2 := L_2\sigma$
9. Menj a 2-re.

Az ítéletlogikához képest a következőképpen módosul a rezolvens fogalma:

2.12. Definíció. (Rezolvens)

Tekintsük a $C_1 = L_1 \vee C_1'$ és $C_2 = L_2 \vee C_2'$ klózokat, ahol L_1 és L_2 azonos alapú és ellentétesen negált literálok, valamint L_1 alapjának és L_2 alapjának létezik legáltalánosabb unifikátora, melyet σ-val fogunk jelölni. A $(C_1' \vee C_2')\sigma$ klózt a C_1 és C_2 *rezolvensének* nevezzük.

Az 1.4. fejezet és az 1.5. fejezet egyéb fogalmai nem módosulnak, vagyis a fenti rezolvens-fogalommal dolgozva[8] tudunk elsőrendű logikában rezolúciós és lineáris inputrezolúciós levezetéseket konstruálni.

2.13. Példa.

A 2.10. példában előállított

$$\mathcal{S} = \left\{ \begin{array}{c} K(c) \\ E(c) \vee I(c) \\ \neg I(y) \vee \neg K(y) \vee E(y) \\ \neg E(z) \vee \neg K(z) \end{array} \right\}$$

klózhalmazról fogjuk most megmutatni, hogy kielégíthetetlen. Ezt *lineáris inputrezolúcióval* próbáljuk bizonyítani. Ám mivel $\mathcal{S}$-ben *nem csak Horn-klózok* vannak ($E(c) \vee I(c)$ nem ilyen), így még ha $\mathcal{S}$ kielégíthetetlen is lenne, sem biztos, hogy le tudjuk belőle vezetni lineáris inputrezolúcióval az üres klózt. Ezért megpróbálhatjuk átalakítani $\mathcal{S}$-t oly módon, hogy csak Horn-klózokat tartalmazzon. Ez könnyen megtehető egy új predikátumszimbólum bevezetésével:

$E'(x)$: „x nem kedvence az elnöknek”

Így átírható $\mathcal{S}$ a következőképpen:

$$\mathcal{S} = \left\{ \begin{array}{c} K(c) \\ \neg E'(c) \vee I(c) \\ \neg I(y) \vee \neg K(y) \vee \neg E'(y) \\ E'(z) \vee \neg K(z) \end{array} \right\}$$

Látható, hogy $\mathcal{S}$ így már csak Horn-klózokat tartalmaz. Adjunk meg tehát $\mathcal{S}$-hez lineáris inputrezolúciós cáfolatot, a következő formában:

[8] Mindehhez társul még az ún. *faktorizáció* művelete, mely nélkül az elsőrendű rezolúció nem teljes. Ennek ismerete nem része a tematikának.

$K(c)$	$E'(z) \vee \neg K(z)$	$\begin{bmatrix} z \\ c \end{bmatrix}$
$E'(c)$	$\neg E'(c) \vee I(c)$	
$I(c)$	$\neg I(y) \vee \neg K(y) \vee \neg E'(y)$	$\begin{bmatrix} y \\ c \end{bmatrix}$
$\neg K(c) \vee \neg E'(c)$	$K(c)$	
$\neg E'(c)$	$E'(z) \vee \neg K(z)$	$\begin{bmatrix} z \\ c \end{bmatrix}$
$\neg K(c)$	$K(c)$	
$\square$		

Az ábra minden sorában a két rezolválandó klóz mellett látható a rezolválás során használt unifikátor is (csak akkor, ha az nem az üres helyettesítés, azaz nem az ϵ).

3. PROLOG

A Prolog logikai programozási nyelv az *elsőrendű lineáris inputrezolúción* alapul. Mivel csak Horn-klózokkal dolgozhatunk, háromféle szintaktikai egységet tudunk Prologban megadni:

1. *Tény (fact)*: pozitív Horn-klóz, azaz egyetlen pozitív literál. Például a $P(x, f(c))$ pozitív Horn-klózt a következő formában tudjuk megadni:

   ```
   p(X,f(c)).
   ```

   Szemléletes jelentése: ez a literál mindig igaz. A Prologban a predikátum- és függvényszimbólumokat kisbetűvel, a változókat nagybetűvel kezdjük.

2. *Szabály (rule)*: vegyes (nem pozitív és nem negatív) Horn-klóz, azaz egy pozitív literált és valahány (nem nulla) negatív literált tartalmaz. Például a $\neg P(c, x) \vee \neg Q(y, f(x)) \vee P(y, x)$ klózt a következő formában tudjuk megadni:

   ```
   p(Y,X) :- p(c,X), q(Y,f(X)).
   ```

   Szemléletes jelentése: ha a szabály jobb oldalának (a szabály *törzsének*) minden literálja igaz, akkor a bal oldalán szereplő literál (a szabály *feje*) is igaz.

3. *Lekérdezés (query)*: negatív Horn-klóz, azaz csupa negatív literálból áll. Például a $\neg P(x) \vee \neg R \vee \neg Q(c, x)$ klózt a következő formában adjuk meg:

   ```
   ?- p(X), r, q(c,X).
   ```

   Szemléletes jelentése: vajon igaz lehet-e egyszerre a lekérdezés mindegyik literálja?

A Prolog program *tények és szabályok* összessége. A lekérdezés a Prolog interpreter promptjához írandó, ezzel indítjuk a lineáris inputrezolúciós levezetést. Ebből következik, hogy a Prolog valamelyest leszűkíti a kezelhető Horn-klózhalmazok körét: a klózhalmaznak pontosan 1 db. negatív klózt (azaz lekérdezést) szabad csak tartalmaznia!

Egy másik plusz megkötést az (eredeti) lineáris inputrezolúcióhoz képest: a levezetés mindig a megadott (egyetlen) *lekérdezéssel indul.* Azaz az 1.19. és a 2.13. példákban az inputrezolúciós levezetés szemléltetésére használt fának a felső centrális (azaz bal felső) klóza mindig a lekérdezés kell legyen!

A harmadik megkötés pedig arra vonatkozik, hogy a levezetés során rezolválásra *milyen sorrendben* választunk ki klózokat (tényeket és szabályokat), illetve hogy egy adott klózon belül milyen sorrendben választjuk a literálokat. A Prolog ezeket a *programban megadott sorrendben* választja ki.

A Prolog levezetési folyamatának szemléltetésére módosítunk az eddig használt fával történő ábrázoláson: a fa csúcsai a centrális klózok lesznek, a mellékklózokat a csúcsokból kiinduló élekre fogjuk írni.

3.1. Példa.

Tekintsük az 1.19. példában megadott

$$\mathcal{S} = \left\{ \begin{array}{c} \neg M \vee \neg K \vee T \\ \neg T \vee \neg R \vee \neg H \\ \neg A' \vee R \\ \neg A' \vee H \\ M \\ K \\ A' \end{array} \right\}$$

input klózhalmazt. Ez a klózhalmaz pontosan 1 db. negatív klózt tartalmaz, azaz tudjuk alkalmazni rá a Prologot. A következő Prolog programot tudjuk felírni:

```
t :- m, k.
r :- a.
h :- a.
m.
k.
a.
```

A lekérdezés pedig a következő lesz:

```
?- t, r, h.
```

A Prolog a következő levezetést generálja:

```
 ?- t, r, h.
        | t :- m, k.
?- r, h, m, k.
        | r :- a.
?- h, m, k, a.
        | h :- a.
 ?- m, k, a.
        | m.
  ?- k, a.
        | k.
   ?- a.
        | a.
        □
```

Mivel sikerült $\mathcal{S}$-ből levezetni az üres klózt, bebizonyítottuk, hogy $\mathcal{S}$ kielégíthetetlen. Ezt a Prolog rendszer egy 'yes' válasszal adja tudtunkra, azaz hogy a lekérdezés (mint a prompt mögé írt literáloknak a konjunkciója) logikai következménye a megadott programnak.

3.2. Példa.

Tekintsük a 2.13. példában megadott

$$\mathcal{S} = \left\{ \begin{array}{c} K(c) \\ \neg E'(c) \vee I(c) \\ \neg I(y) \vee \neg K(y) \vee \neg E'(y) \\ E'(z) \vee \neg K(z) \end{array} \right\}$$

input klózhalmazt. Ez a klózhalmaz is pontosan 1 db. negatív klózt tartalmaz. A következő Prolog programot tudjuk felírni:

```
k(c).
i(c) :- e(c).
e(Z) :- k(Z).
```

A lekérdezés pedig a következő lesz:

```
?- i(Y), k(Y), e(Y).
```

A Prolog a következő levezetést generálja:

```
?- i(Y), k(Y), e(Y).
           | i(c) :- e(c).
   ?- k(c), e(c).
           | k(c).
      ?- e(c).
           | e(Z) :- k(Z).
      ?- k(c).
           | k(c).
           □
```

Vegyük észre, hogy a levezetés során többször is alkalmaztuk az unifikációt, azaz az `Y` és `Z` változókat termekkel helyettesítettük. Mivel `Y` a lekérdezésben is előfordult, így a Prolog rendszer most nem '`yes`' válasszal tér vissza, hanem az '`Y = c`' válasszal.

Amennyiben a levezetés adott ágán az üres klóz nem vezethető le, a Prolog *backtracking* módszerrel visszalépked a korábbi csúcsokba, és ezekből más irányban (a csúcsban található klóz első literálját más ténnyel vagy más szabály fejével unifikálva) próbál elindulni.