Programmiersprachen

Logik im Fingerhut

Berthold Hoffmann

Studiengang Informatik
Universität Bremen

Wintersemester 2004/2005
(Vorlesung am 24. Januar 2005)

Wozu formale Logik?

- präzise Beschreibung von Aussagen über "die Welt" bzw. über "verschiedene Welten"
- Ziehen und Überprüfen von Schlussfolgerungen
- "maschinelles Beweisen" (*automated theorem proving*)
- Anwendung: Darstellung von Wissen in der Künstlichen Intelligenz

Logik im Fingerhut (24. Januar 2005)

1. Aussagenlogik
   - Syntax
   - Semantik
   - Normalformen
   - Resolution
2. Prädikatenlogik
3. Hornklauseln

Formeln

- $\mathcal{A} = \{A_1, A_2, \cdots\}$ ist eine Menge von "atomaren Formeln" oder "Aussagen"
- Formeln sind Boole'sche Ausdrücke über Aussagen

$$\mathcal{F} ::= A_1 \mid A_2 \mid \cdots \mid \neg\mathcal{F} \mid \mathcal{F} \wedge \mathcal{F} \mid \mathcal{F} \vee \mathcal{F}$$

- Abkürzungen:

$$\begin{array}{rcll} A \supset B & := & \neg A \vee B & \text{(Implikation)} \\ A \leftrightarrow B & := & (A \supset B) \wedge (B \supset A) & \text{(Äquivalenz 1)} \\ (& := & (A \wedge B) \vee (\neg A \wedge \neg B) \ ) & \text{(Äquivalenz 2)} \end{array}$$

Belegung und Interpretation

- Die Menge $\mathcal{W} = \{0, 1\}$ definiert Wahrheitswerte
- Eine Abbildung $\alpha\colon \mathcal{A} \to \mathcal{W}$ belegt Aussagen mit Wahrheitswerten
- Ihre Fortsetzung $\alpha^*\colon \mathcal{F} \to \mathcal{W}$ interpretiert Formeln

$$\begin{array}{rcl} \alpha^*(A_i) & = & \alpha(A_i) \text{ für alle } A_i \in \mathcal{A} \\ \alpha^*(\neg F) & = & \begin{cases} 1 & \text{wenn } \alpha^*(F) = 0 \\ 0 & \text{sonst} \end{cases} \\ \alpha^*(F \wedge G) & = & \begin{cases} 1 & \text{wenn } \alpha^*(F) = 1 \text{ und } \alpha^*(G) = 1 \\ 0 & \text{sonst} \end{cases} \\ \alpha^*(F \vee G) & = & \begin{cases} 1 & \text{wenn } \alpha^*(F) = 1 \text{ oder } \alpha^*(G) = 1 \\ 0 & \text{sonst} \end{cases} \end{array}$$

Erfüllbarkeit und Gültigkeit

- α erfüllt F (α ist Modell für F, $\alpha \models F$) wenn $\alpha^*(F) = 1$
- Eine Formel F heißt erfüllbar wenn sie ein Modell hat.
- Eine Formel F heißt gültig (eine Tautologie). wenn alle Belegungen α Modell für F sind ($\models F$).
- F und G heißen äquivalent (geschrieben $F \equiv G$) wenn $\alpha \models F$ *gdw.* $\alpha \models F$.
- Aus F folgert G wenn alle alle Modelle für F auch Modelle für G sind.

Lemma

- *F ist gültig gdw. $\neg F$ nicht erfüllbar ist.*
- *$F \equiv G$ gdw. $F \leftrightarrow G$ gültig ist.*
- *Aus F folgt G gdw. $F \supset G$ gültig ist.*

Aussagenlogische Gesetze

Satz (Aussagenlogische Äquivalenzen für "$\wedge$")

$$\begin{array}{rcll} F \wedge F & \equiv & F & \textit{Idempotenz} \\ F \wedge G & \equiv & G \wedge F & \textit{Kommutativitiät} \\ F \wedge (G \wedge H) & \equiv & (F \wedge G) \wedge H & \textit{Assoziativität} \\ F \wedge (F \vee G) & \equiv & F & \textit{Absorbtion} \\ F \wedge (G \vee H) & \equiv & (F \wedge G) \vee (F \wedge H) & \textit{Distributivität} \\ \neg(F \wedge G) & \equiv & \neg F \vee \neg G & \textit{de Morgan} \\ F \wedge G & \equiv & G \text{ falls } \models F & \textit{Tautologie} \\ F \wedge G & \equiv & F \text{ falls } \not\models F & \textit{Unerfüllbarkeit} \end{array}$$

Aussagenlogische Gesetze

Satz (Aussagenlogische Äquivalenzen für "$\vee$")

$$\begin{array}{rcll} F \vee F & \equiv & F & \textit{Idempotenz} \\ F \vee G & \equiv & G \vee F & \textit{Kommutativitiät} \\ F \vee (G \vee H) & \equiv & (F \vee G) \vee H & \textit{Assoziativität} \\ F \vee (F \wedge G) & \equiv & F & \textit{Absorbtion} \\ F \vee (G \wedge H) & \equiv & (F \vee G) \wedge (F \vee H) & \textit{Distributivität} \\ \neg(F \vee G) & \equiv & \neg F \wedge \neg G & \textit{de Morgan} \\ F \vee G & \equiv & F \text{ falls } \models F & \textit{Tautologie} \\ F \vee G & \equiv & G \text{ falls } \not\models F & \textit{Unerfüllbarkeit} \end{array}$$

Prinzip (Dualität)

Vertauschen von "$\wedge$" und "$\vee$" in Formeln bewahrt Äquivalenz.

Äquivalentes Umformen

Definition (Substitution)

$H[A/F]$ bezeichnet das Ersetzen aller Teilformeln A in H durch F.

Satz (Ersetzbarkeit)

Sei H eine Formel, in der die Aussage A vorkommt. Dann folgt aus $F \equiv G$, dass $H[A/F] \equiv H[A/G]$.

Korollar (Korrektheit und Vollständigkeit der Gesetze)

Eine Formel F ist

- *gültig gdw. $F \equiv \cdots \equiv A \vee \neg A$.*
- *unerfüllbar gdw. $F \equiv \cdots \equiv A \wedge \neg A$.*

(Hierbei ist A eine beliebige Aussage.)

Äquivalentes Umformen am Beispiel

Beispiel (Äquivalenz der Definitionen von "$\leftrightarrow$")

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

Normalformen aussagenlogischer Formeln

- Eine atomare Formel oder eine negierte atomare Formel heißt Literal.
- F ist in konjunktiver Normalform (**KNF**) wenn

$$F = (L_{1,1} \vee \cdots \vee L_{1,m_1}) \wedge \cdots \wedge (L_{n,1} \vee \cdots \vee L_{n,m_n})$$

wobei die $L_{i,j}$ Literale sind $(1 \leqslant j \leqslant m_i, 1 \leqslant i \leqslant n)$

Satz

Jede Formel F besitzt eine äquivalente **KNF**.

Beweis.

Äquivalentes Umformen nach Gesetzen der Boole'schen Algebra (z. B. $\neg(F \vee G) = \neg F \wedge \neg G$ und $\neg\neg F = F$). □

KNF von aussagenlogischen Formeln

Beispiel

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

Aussagenlogische Resolventen

$\bar{L}$ bezeichnet das duale Literal $\bar{L} = \begin{cases} \neg A_i & \text{wenn } L = A_i \\ A_i & \text{wenn } L = \neg A_i \end{cases}$

Definition

R ist Resolvente von zwei Klauseln K_1, K_2, wenn

1. es gibt ein Literal L mit $L \in K_1$ und $\bar{L} \in K_2$
2. $R = K_1 \cup K_2 \setminus \{L, \bar{L}\}$

Lemma (Resolutionslemma der Aussagenlogik)

Für jede Klauselmenge F mit Klauseln K_1 und K_2 und jede Resolvente R von K_1 und K_2 sind F und $F \cup R$ äquivalent.

Aussagenlogische Resolution

Definition (Resolventen)

$$
\begin{aligned}
\mathrm{Res}(F) &= F \cup \{R \mid R \text{ ist Resolvent zweier Klauseln in } F\} \\
\mathrm{Res}^0(F) &= F \\
\mathrm{Res}^{n+1}(F) &= \mathrm{Res}(\mathrm{Res}^n(F)) \\
\mathrm{Res}^*(F) &= \bigcup_{n \geqslant 0} \mathrm{Res}^n(F)
\end{aligned}
$$

Lemma (Termination)

Für jede endliche Klauselmenge F gilt $Res^k(F) = Res^{k+1}(F) = \cdots = Res^*(F)$ für irgendein $k \geqslant 0$.

Aussagenlogischer Resolutionsalgorithmus

Satz

F ist unerfüllbar gdw. $\square \in Res^*(F)$.

Korollar

Unerfüllbarkeit aussagenlogischer Formeln ist entscheidbar.

Algorithmus (Resolution)

Eingabe: Eine Klauselmenge F

repeat $G := F$;
$F := Res(F)$;
until $(\square \in F)$ **or** $(F = G)$;
if $\square \in F$ **then** *"F ist unerfüllbar"*
else *"F ist erfüllbar"*

Aussagenlogische Resolution

Beispiel (Resolution)

$\{A, B, \neg C\}$ $\{A, B, C\}$ $\{A, \neg B\}$ $\{\neg A\}$

$\{A, B\}$

$\{A\}$

$\square$

Logik im Fingerhut (24. Januar 2005)

1. Aussagenlogik
2. Prädikatenlogik
   - Syntax
   - Semantik
   - Normalformen
   - Resolution
3. Hornklauseln

Erweiterungen der Aussagenlogik

- Atomare Formeln ("Aussagen") sind Prädikate über Termen.
- Terme sind Ausdrücke über einem Bereich (*domain*).
- Formeln können über Individuen des Bereichs quantifiziert werden.

Terme

- Symbole
  - Eine Menge X von Individuen-Variablen $x_1, x_2, \ldots$
  - Eine Familie $F = \bigcup_{k \geqslant 0} F^k$ nach Stelligkeit sortierter Funktionssymbole $f_1, f_2, \ldots$.
  - Eine Familie $P = \bigcup_{k \geqslant 0} P^k$ nach Stelligkeit sortierter Prädikatsymbole $p_1, p_2, \ldots$.
  - F und P enthalten endlich viele Symbole.
- Terme

$$\begin{array}{rl} x \in \mathcal{T} & \text{wenn } x \in X \\ f(t_1, \ldots, t_k) \in \mathcal{T} & \text{wenn } f \in F^k \text{ und } t_i \in \mathcal{T}, 1 \geqslant i \geqslant k\} \end{array}$$

(Wenn $f \in F^0$, ist $f \in \mathcal{T}$.)

prädikatenlogische Formeln

- Atomare Formeln

$$\mathcal{A} = \{p(t_1, \ldots, t_k) \mid p \in P^k \text{ und } t_i \in \mathcal{T}, 1 \geqslant i \geqslant k\}$$

- Allgemeine Formeln

$$\begin{array}{rl} & \mathcal{A} \subseteq \mathcal{F} \\ F \vee G, F \wedge G \in \mathcal{F}, & \text{wenn } F, G \in \mathcal{F} \\ \neg F \in \mathcal{F}, & \text{wenn } F \in \mathcal{F} \\ \exists x F, \forall x F, & \text{wenn } x \in X \text{ und } F \in \mathcal{F} \end{array}$$

- Formeln erster Stufe: Variablen bezeichnen Individuen, keine Funktionen oder Prädikate.

Belegung, Interpretation atomarer Formeln

- Eine Belegung α besteht aus einer nichtleeren Menge U (dem Universum) und trifft folgende Zuordnungen:
  - Für alle $x \in X$ ein Element $u \in U$.
  - Für alle $f \in F^k$ eine Funktion $f_\alpha \colon U^k \to U$.
  - Für alle $p \in P^k$ eine Funktion $p_\alpha \colon U^k \to \mathcal{W}$. ($\mathcal{W} = \{0, 1\}$ sind die Wahrheitswerte.)
- Die Fortsetzung $\alpha^* \colon \mathcal{A} \to \mathcal{W}$ wertet atomare Formeln aus

$$\begin{array}{rcl} \alpha^*(x) & = & \alpha(x) \text{ für alle } x \in X \\ \alpha^*(f(t_1, \ldots t_k)) & = & f_\alpha(\alpha^*(t_1), \ldots, \alpha^*(t_k)) \\ \alpha^*(p(t_1, \ldots t_k)) & = & p_\alpha(\alpha^*(t_1), \ldots, \alpha^*(t_k))) \end{array}$$

für alle $f \in F^k$ bzw. $p \in P^k$ und $t_i \in \mathcal{T}, 1 \geqslant i \geqslant k$

Interpretation von Formeln

- Die Fortsetzung $\alpha^* \colon \mathcal{F} \to \mathcal{W}$ interpretiert Formeln

$$\begin{array}{rcl} \alpha^*(A) & = & \alpha(A_i) \text{ für alle } A \in \mathcal{A} \\ \alpha^*(\neg F) & & \\ \alpha^*(F \wedge G) & \} & \text{wie in der Aussagenlogik} \\ \alpha^*(F \vee G) & & \\ \alpha^*(\exists x F) & = & \begin{cases} 1 & \text{wenn } \alpha^*(F) = 1 \text{ für mind. ein } \alpha \\ 0 & \text{sonst} \end{cases} \\ \alpha^*(\forall x F) & = & \begin{cases} 1 & \text{wenn } \alpha^*(F) = 1 \text{ für alle } \alpha \\ 0 & \text{sonst} \end{cases} \end{array}$$

Erfüllbarkeit und Gültigkeit

- α erfüllt F (α ist Modell für F, $\alpha \models F$) wenn $\alpha^*(F) = 1$
- Eine Formel F heißt erfüllbar wenn sie ein Modell hat.
- Eine Formel F heißt gültig (eine Tautologie). wenn alle Belegungen α Modell für F sind ($\models F$).
- F und G heißen äquivalent (geschrieben $F \equiv G$) wenn $\alpha \models F$ *gdw.* $\alpha \models F$.
- Aus F folgert G wenn alle alle Modelle für F auch Modelle für G sind.

Normalformen der Prädikatenlogik

Lemma (Bereinigung)

Jede Formel F hat eine äquivalente bereinigte Form G so dass

- *alle quantifizieren Variablen sind verschieden*
- *quantifizierte und freie Variablen sind verchieden*

Lemma (Existenz-Abschluss)

Eine Formel F mit freien Variablen $x_1, \ldots, x_n$ ist erfüllbar gdw. $\exists x_1 \ldots \exists x_n F$ erfüllbar ist.

Lemma (Pränexform)

Jeder Formel F gibt es eine äquivalente Pränexform G
$F = \Diamond_1 y_1 \ldots \Diamond_n y_n F'$
wobei $\Diamond_i \in \{\exists, \forall\}$ und F' quantorenfrei ist.

Skolem-Normalform

Algorithmus (Skolemisierung)

Eingabe: Eine bereinigte abgeschlossene Formel $F \in \mathcal{F}$ in Pränexform.

while *F enthält Existenzquantor*
do begin
Sei $F = \forall x_1 \ldots \forall x_n G$, $n \geqslant 0$;
Sei f ein frisches Funktionssymbol aus F^n;
$F := \forall x_1 \ldots \forall x_n G[x/f(x_1, \ldots, x_n)]$
end

Lemma

Eine bereinigte abgeschlossene Formel $F \in \mathcal{F}$ in Pränexform ist erfüllbar gdw. ihre Skolemform erfüllbar ist.

Klauselform der Prädikatenlogik

Algorithmus (Bestimmen der Klauselform)

Eingabe: Eine prädikatenlogische Formel $F \in \mathcal{F}$.
Ausgabe: Eine erfüllbarkeits-äquivalente Klauselmenge

1. *Bereinigen liefert F_1 äquivalent zu F.*
2. *Abschluss liefert F_2 erfüllbarkeitsäquivalent zu F_1.*
3. *Die Pränexnormalform F_3 ist äquivalent zu F_2.*
4. *Die Skolemform F_4 ist erfüllbarkeitsäquivalent zu F_3.*
5. *Schreibe F_4 ohne Quantoren als Klauselmenge*

Lemma

Eine bereinigte abgeschlossene Formel $F \in \mathcal{F}$ in Pränexform ist erfüllbar gdw. ihre Skolemform erfüllbar ist.

Herbrand-Expansion

Sei $F = \forall x_1 \ldots \forall x_n F^*$ in Skolemform

Definition (Freie Termalgebra, Herbrand-Universum)

1. Alle nullstelligen Funktionssymbol $a \in F^0$ aus F sind in U_F (aber mindestens eines).
2. $f(t_1, \ldots, t_k) \in U_F$ falls $f \in F^k$ und $t_1, \ldots, t_k$ in U_F

Definition (Herbrand-Expansion)

$$E(F) = \{F^*[x_1/t_1] \cdots [x_n/t_n] \mid t_1, \ldots, t_k \in U_F\}$$

Satz

Eine Aussage F in Skolemform ist genau dann erfüllbar wenn $E(F)$ erfüllbar ist (im aussagenlogischen Sinn)

Semi-Entscheidbarkeit der Unerfüllbarkeit

Satz (Herbrand)

Eine Skolemform F ist unerfülbar gdw. eine endliche Teilmenge von $E(F)$ unerfüllbar ist (im aussagenlogischen Sinn).

Algorithmus (Verfahren von Gilmore)

Eingabe: Eine Formel F in Skolemform.

Sei $E(F) = \{F_1, F_2, \ldots\}$
$n := 0$;
repeat $n := n + 1$
until $\{F_1, F_2, \ldots, F_n\}$ *ist unerfüllbar.*
"F ist unerfüllbar"; **halt**

Unifikation

Definition

- Eine Substitution σ ist eine Menge von Paaren $[x_i/t_i]$ mit $x_i \in X$ und $t_i \in U_F$.
- F^σ ist die Formel, die durch Anwenden von σ auf F entsteht.
- σ ist Unifikator einer Menge $\mathbf{L} = \{L_1, L_2, \ldots, L_k\}$ von Literalen wenn
  $L_1^\sigma = L_2^\sigma = \cdots = L_k^\sigma$
  (Dann heißt $\mathbf{L}$ unifizierbar.)
- Ein Unifikator σ von $\mathbf{L}$ ist der allgemeinste Unifikator wenn es für alle Unifikatoren τ' von $\mathbf{L}$ eine Substitution τ' mit $\sigma\tau' = \tau$ gibt.

Unifikation

Satz (Robinson)

Unifizierbare Literale haben immer einen allgemeinsten Unifikator.
(Er ist bis auf Variablennamen eindeutig.)

Unifikations-Algorithmus

Algorithmus (Allgemeinster Unifikator)

Eingabe: Nicht-leere Literalmenge $\mathbf{L} = \{L_1, l_2, \ldots\}$
$\sigma \leftarrow \varepsilon$; *– die leere Substitution*
while $|\mathbf{L}\sigma| > 1$
do *Finde die ersten verschiedenen Zeichen in $L_1, L_2 \in \mathbf{L}\sigma$*
if *kein Zeichen ist eine Variable*
then *"Nicht unifizierbar"*; **halt**
else *Sei oBdA $z \in X$, und t der mit z' beginnende Term*
if *z taucht in t auf* **then** *"Nicht unifizierbar"*; **halt**
else $\sigma \leftarrow \sigma[x/t]$
end if
end if
end do

prädikatenlogische Resolventen

$\bar{L}$ bezeichnet das duale Literal

$$\bar{L} = \begin{cases} \neg p(\cdots) & \text{wenn } L = p(\cdots) \\ p(\cdots) & \text{wenn } L = \neg p(\cdots) \end{cases}$$

Definition

R ist Resolvente von zwei Klauseln K_1, K_2, wenn

1. K_1 und K_2 enthalten keine gemeinsamen Variablen
2. es gibt $\{L_1, \ldots L_m\} \subseteq K_1$ und $\{L'_1, \ldots L'_n\} \subseteq K_2$ so dass $\mathbf{L} = \{\bar{L}_1, \ldots, \bar{L}_m, L'_1, \ldots, L'_n\}$ einen allgemeinsten Unifikator σ hat.
3. $R = ((K_1 \setminus \{L_1, \ldots L_m\} \cup (K_2 \setminus \{L'_1, \ldots L'_n\}))^\sigma$

prädikatenlogische Resolution

Definition (prädikatenlogische Resolvente)

Sei F eine Klauselmenge.
$\mathrm{Res}(F) = F \cup \{R \mid R \text{ ist Resolvent zweier Klauseln in } F\}$

$$\begin{aligned} \mathrm{Res}^0(F) &= F \\ \mathrm{Res}^{n+1}(F) &= \mathrm{Res}(\mathrm{Res}^n(F)) \\ \mathrm{Res}^*(F) &= \bigcup_{n \geqslant 0} \mathrm{Res}^n(F) \end{aligned}$$

Satz

F ist unerfüllbar gdw. $\square \in Res^*(F)$.

Logik im Fingerhut (24. Januar 2005)

1. Aussagenlogik
2. Prädikatenlogik
3. Hornklauseln

Hornformeln

Definition (Hornklausel)

- Eine Klausel mit höchstens einem positiven Literal heißt Hornklausel.
  - kein positives Literal $\Longrightarrow$ Zielklausel.
  - ein positives Literal $\Longrightarrow$ Programmklausel.
- Eine Menge von Hornklauseln heißt Hornformel.

Lineare Resolution

Definition (Lineare Resolution)

- *Eingabe*: Eine Zielklauseln und beliebig viele Prgrammklauseln
- Jeder Schritt resolviert die Zielklausel mit einer Programmklausel

Satz

Lineare Resolution ist vollständig für Hornklauseln.

Weshalb Hornklauseln?

- Viele mathematische Theorien sind so axiomatisierbar.
- Berechnung ist "vollständig".
- Berechnung ist "effektiv".

Interpretationen von Hornklauseln

- Zielklauseln sind Existenz-Abfragen
$$\begin{aligned} & \{\neg G_1, \ldots, \neg G_n\} \\ \Leftrightarrow \quad & \forall x_1 \cdots x_n \neg G_1 \vee \cdots \vee \neg G_n \\ \Leftrightarrow \quad & \forall x_1 \cdots x_n \neg (G_1 \wedge \cdots \wedge G_n) \\ \Leftrightarrow \quad & \neg (\exists x_1 \cdots x_n (G_1 \wedge \cdots \wedge G_n)) \end{aligned}$$
- Programmklauseln sind Schlussregeln.
$$\begin{aligned} & \{P, \neg Q_1, \ldots, \neg Q_n\} \\ \Leftrightarrow \quad & \forall x_1 \cdots x_n (P \vee \neg Q_1 \vee \cdots \vee \neg Q_n) \\ \Leftrightarrow \quad & \forall x_1 \cdots x_n (P \vee \neg (Q_1 \wedge \cdots \wedge Q_n)) \\ \Leftrightarrow \quad & \forall x_1 \cdots x_n (P \leftarrow (Q_1 \wedge \cdots \wedge Q_n)) \end{aligned}$$
- Einelementige Programmklauseln definieren Fakten.
$\{P\} \Leftrightarrow \forall x_1 \cdots x_n P$

Interpretationen von Hornklauseln

- Lineare Resolution widerlegt die Anahme $\neg G \wedge P$.
- Die allgemeinsten Unifikatoren bestimmen Gegenbeispiele (Substitutionen der Variablen, die G erfüllen)

Logik im Fingerhut (24. Januar 2005)

1. Aussagenlogik
2. Prädikatenlogik
3. Hornklauseln

Zusammenfassung

- Prädikatenlogik formalisiert Modelle der realen Welt
- Beschränkung macht sie "rechnertauglich"
  - Formeln erster Stufe
  - Skolem-Normalform, Hornklauseln
  - Resolution als einzige Inferenzregel

Nächstes Mal

- Umsetzung von Hornformeln in PROLOG
- Nicht-logische Konzepte von PROLOG
- Andere logische und funktional-logische Sprachen

Nachlese(n)


R. Kowalski.
Algorithm = Logic + Control.
Journal ACM 22: 424–436, 1979.


Uwe Schöning.
Logik für Informatiker (5. Aufl.).
Hochschultaschenbuch, Spektrum Akademischer Verlag, Heidelberg–Berlin, 2000.