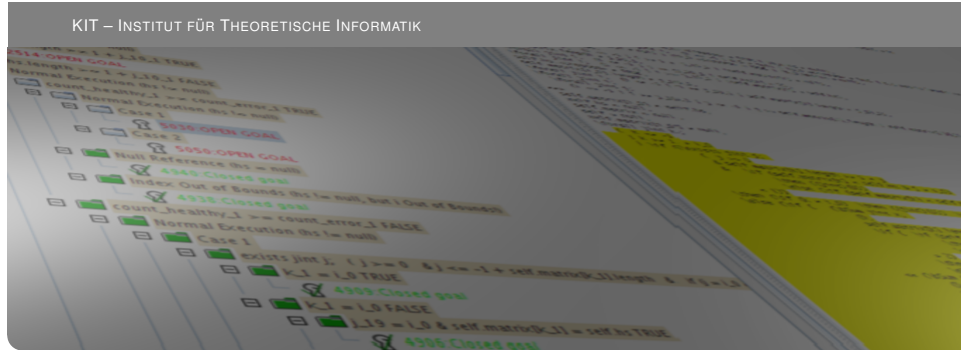


Formale Systeme

Prof. Dr. Bernhard Beckert, WS 2017/2018

Wiederholung

KIT – INSTITUT FÜR THEORETISCHE INFORMATIK



- ▶ Aussagenlogik
- ▶ Prädikatenlogik
- ▶ Reduktionssysteme
- ▶ Modale Logik
- ▶ LTL, Büchi Automaten, Modellprüfung (model checking)

Aussagenlogik

Was Sie wissen sollten

Aussagenlogik, Syntax und Semantik

- ▶ Definitionen der logischen Grundbegriffe beherrschen
Erfüllbarkeit, Allgemeingültigkeit, $M \models A$,
Boolesche Funktion f_A einer Formel A , etc
- ▶ Definition der verschiedenen Normalformen
disjunktive, konjunktive Normalform, Negationsnormalform,
kurze konjunktive Normalform, Shannon Normalform,
(reduzierte) Shannongraphen.
- ▶ Craigscher Interpolationssatz
- ▶ Einfache aussagenlogische Tautologien erkennen.
- ▶ Gegebene Formeln in Normalformen transformieren.
- ▶ Shannongraphen zu Boolesche Funktionen konstruieren
und umgekehrt.

Was Sie wissen sollten

SAT solver

- ▶ Davis-Putnam-Logemann-Loveland Verfahren (DPLL)
- ▶ Für einfache Formeln mit Hilfe des DPLL Erfüllbarkeit bzw. Unerfüllbarkeit feststellen.
- ▶ Einfache Anwendungen durch aussagenlogische Formeln formalisieren können.

Prädikatenlogik

Was Sie wissen sollten

Prädikatenlogik, Syntax und Semantik

- ▶ Definition der Formeln und der Strukturen für PK1
- ▶ Begriff der Substitution kennen und anwenden können.
- ▶ Den Begriff *Unifikation* kennen und den allgemeinsten Unifikator einfacher Terme berechnen können.
- ▶ Die logischen Grundbegriffe beherrschen
- ▶ Einfache semantische Tautologien erkennen können
- ▶ Einfache Sachverhalte in PL1 formalisieren können
- ▶ Definition der verschiedenen Normalformen
- ▶ Gegebene Formeln in Normalformen transformieren
- ▶ Die Aussage des Substitutionslemmas kennen
- ▶ Den Satz von Herbrand kennen

Signatur $\Sigma = (F_\Sigma, P_\Sigma, \alpha_\Sigma)$

- ▶ F_Σ Menge aller Funktionszeichen, 0-stellige Funktionsymbole heißen Konstantensymbole, z.B. 0 als Konstante für die kleinste natürliche Zahl.
- ▶ P_Σ Menge aller Prädikatszeichen.

Menge aller Formeln For_Σ

1. $\{\mathbf{1}, \mathbf{0}\} \cup At_\Sigma \subseteq For_\Sigma$
2. Mit $x \in Var$ und $A, B \in For_\Sigma$ sind ebenfalls in For_Σ :
 $\neg A, (A \wedge B), (A \vee B), (A \rightarrow B), (A \leftrightarrow B), \forall xA, \exists xA$

1 kann als Abkürzung für $A \vee \neg A$

0 kann als Abkürzung für $A \wedge \neg A$

aufgefasst werden.

Semantik der Prädikatenlogik

Signatur $\Sigma_{arith} = \{+, *, \leq\}$

Die mathematischen ganzen Zahlen

$$\mathcal{Z} = (\mathbb{Z}, +_{\mathcal{Z}}, *_{\mathcal{Z}}, \leq_{\mathcal{Z}}).$$

Die ganzen Zahlen in Java

$$\mathcal{Z}_{Jint} = (\mathbb{Z}_{Jint}, +_{Jint}, *_{Jint}, \leq_{Jint}).$$

wobei:

$$\mathbb{Z}_{Jint} = [minInt, maxInt] = [-2147483648, 2147483647]$$

$$n +_{Jint} m = \text{Java Semantik}$$

$$n *_{Jint} m = \text{Java Semantik}$$

$$n \leq_{Jint} m \Leftrightarrow n \leq_{\mathcal{Z}} m$$

Formel ϕ	$\mathcal{Z} \models \phi$	$\mathcal{Z}_{jint} \models \phi$
$\forall x \exists y (x < y)$	ja	nein
$\forall x \forall y ((x + 1) * y = x * y + y)$	ja	ja
$\exists x (0 < x \wedge x + 1 < 0)$	nein	ja

Prädikatenlogik

Tautologien erkennen

- (a) $\phi_a = \exists x \neg(\forall x(f(x) \doteq f(x)))$
- (b) $\phi_b = \forall x(f(x) \doteq c) \rightarrow f(f(f(c))) \doteq c$
- (c) $\phi_c = \forall x(\forall y(p(x) \vee \neg p(y)))$
- (d) $\phi_e = ((r \rightarrow s) \rightarrow r) \rightarrow (r \rightarrow (s \rightarrow r))$

Bemerkung: p, q, r, s sind Prädikatssymbole, f, g Funktionssymbole (jeweils mit der richtigen Stelligkeit), c ein Konstantensymbol (nullstelliges Funktionssymbol) und x, y sind Variablen. Eine Formel kann mehr als eine der genannten Eigenschaften haben.

Wir haben bewiesen

$$\begin{aligned} & \forall x \forall y \forall z (r(x, y) \wedge r(y, z) \rightarrow r(x, z)) \wedge \\ & \forall x \forall y (r(x, y) \rightarrow r(y, x)) \wedge \\ & \forall x \exists y (r(x, y)) \end{aligned} \quad \rightarrow \quad \forall x (r(x, x))$$

Gilt auch

$$\begin{aligned} & \forall x \forall y \forall z (r(x, y) \wedge r(y, z) \rightarrow r(x, z)) \wedge \\ & \forall x \exists y (r(x, y)) \wedge \\ & \forall x (r(x, x)) \end{aligned} \quad \rightarrow \quad \forall x \forall y (r(x, y) \rightarrow r(y, x))$$

Wie sieht ein mögliches Gegenbeispiel aus?

Herbrand Struktur

Eine Interpretation (D, I) heißt *Herbrand-Struktur*, wenn

1. $D = \text{Term}_{\Sigma}^0 =$ Menge der Grundterme.
2. $I(f)(t_1, \dots, t_n) = f(t_1, \dots, t_n)$
für alle Funktionssymbole $f \in \Sigma$
und beliebige Grundterme t_1, \dots, t_n .

Satz von Herbrand

Σ enthalte mindestens eine Konstante, und es sei M eine Menge geschlossener, universell quantifizierter Formeln. Ferner enthalte keine Formel in M das Gleichheitssymbol \doteq .

M hat ein Modell $\Rightarrow M$ hat ein Herbrand-Modell

Was Sie wissen sollten

Beweistheorie

- ▶ Die Ziele der Beweistheorie verstehen.
- ▶ Grundidee des Hilbertkalküls verstehen.
- ▶ Resolutionskalkül, Tableaukalkül kennen.
- ▶ Beweisidee für den Korrektheits- und Vollständigkeitsbeweis des aussagenlogischen Resolutionskalküls kennen.
- ▶ Beweisidee für den Korrektheits- und Vollständigkeitsbeweis des Tableaukalküls kennen.
- ▶ für kleine Beispiele Ableitungen im Resolutionskalkül und Tableaukalkül, für Aussagen- und Prädikatenlogik, finden können
- ▶ Aussagenlogische Tableauregeln aus Wahrheitstafeln konstruieren.

PEANO ARITHMETIK

- ▶ Grundidee der Peano Arithmetik kennen.
- ▶ Entscheidbarkeitsresultate zur Peano Arithmetik kennen.

Resultate

1. $Th(\mathcal{N})$ ist nicht rekursiv.
2. $Th(\mathcal{N})$ ist nicht rekursiv aufzählbar
3. $Cn(PA) \subsetneq Th(\mathcal{N})$
4. $Th(\langle \mathbb{N}, +, 0, 1 \rangle)$ ist rekursiv (entscheidbar).
Presburger Arithmetik

Kann man nicht jede Multiplikation durch Additionen ersetzen?

z.B. $3 \cdot 5 = 5 + 5 + 5$

Da funktioniert nicht bei der folgenden Frage:

$$\mathcal{N} \models \forall x \forall y (x^2 - 9 * y^2 \doteq 1 \rightarrow (x \doteq 1 \vee x \doteq -1) \wedge y \doteq 0)?$$

JML

- ▶ grundlegende Konzepte von JML kennen
- ▶ einfache JML Spezifikationen lesen und erklären können
- ▶ einfache Spezifikationen in JML formalisieren können

Reduktionssysteme

- ▶ Wichtigsten Eigenschaften von Reduktionssystemen und die Zusammenhänge zwischen ihnen kennen
- ▶ Gerichtete Termersetzungssysteme kennen und kleine Beispiele rechnen können.

Modale Logik

- ▶ Definition von Syntax und Semantik (Kripke Strukturen) beherrschen.
- ▶ Allgemeingültige Formeln erkennen können, auch für Allgemeingültigkeit relativ zu Kripke Strukturen mit Einschränkungen and die Zugänglichkeitsrelation R .
- ▶ Für einfache Eigenschaften von R charakterisierende Formeln finden.

Das Symbol \models ist überladen.

1. $M \models A$ bedeutet

Die Formel A ist eine logische Folgerung aus der Formelmenge M .

2. $(\mathcal{K}, s) \models A$ bedeutet

Die Formel A ist wahr im Zustand s der Kripke-Struktur \mathcal{K} .

Das Symbol \vdash steht dagegen für die Ableitbarkeit in einem Kalkül. Zum Beispiel bedeutet $\vdash_{\text{H0}} A$

Die Formel A ist im aussagenlogischen Hilbertkalkül ohne Voraussetzungen ableitbar

Nach dem Beweis des jeweiligen Vollständigkeitssatzes wissen wir, daß \models und \vdash zusammenfallen.

LTL

- ▶ LTL-Formeln lesen können
- ▶ Einfache temporale Eigenschaften in LTL formalisieren können.
- ▶ Zusammenhang zwischen LTL und Büchi Automaten kennen.
- ▶ Konzept der LTL Modellprüfung kennen.

$$\begin{array}{lll} \dots & \dots & \dots \\ \xi \models \Box A & \text{gdw} & \text{für alle } n \in \mathbb{N} \text{ gilt } \xi_n \models A \\ \xi \models \Diamond A & \text{gdw} & \text{es gibt ein } n \in \mathbb{N} \text{ mit } \xi_n \models A \\ \dots & \dots & \dots \end{array}$$

Wobei ξ_n für das bei n beginnende Endstück von ξ steht, i.e.
 $\xi_n(m) = \xi(n + m)$.

Alternative

$$\begin{array}{lll} \dots & \dots & \dots \\ (\xi, i) \models \Box A & \text{gdw} & \text{für alle } j \in \mathbb{N} \text{ mit } i \leq j \text{ gilt } (\xi, j) \models A \\ (\xi, i) \models \Diamond A & \text{gdw} & \text{es gibt ein } j \in \mathbb{N} \text{ mit } i \leq j \text{ und } (\xi, j) \models A \\ \dots & \dots & \dots \end{array}$$

$$p \vee q \leftrightarrow q \wedge (p \vee q)$$

Allgemeines

Häufig gestellte Frage

Ist es erlaubt, jede einigermaßen verbreitete Notation zu verwenden, solange man an einem aktuellen Informatik Lehrbuch nachweisen kann, daß es diese Notation gibt?

Antwort

Bitte, halten Sie sich an die in der Vorlesung und im Skriptum verwendete Notation.

Wir werden bei der Korrektur der Klausur versuchen flexibel zu sein. Das heißt aber nicht, daß wir alles akzeptieren können.

Häufig gestellte Frage

Werden in der Klausur Definitionen abgefragt?

Antwort

In der Regel werden Definitionen nicht direkt abgefragt, sondern Aufgaben gestellt, die voraussetzen, daß man die Definition kennt.

Sie können mit Ankreuzaufgaben im Wert ungefähr 10 Punkten aus 60 rechnen.

Häufig gestellte Frage

Wie kann man Prädikaten- und Funktionssymbole in einer Aufgabenstellung auseinanderhalten?

Antwort

In den meisten Fällen ist das explizit in der Signatur Σ vorgegeben.

Diese Information kann man in den meisten Fällen auch erschließen.

$$(f(a, g(y)) \wedge u(p(x), q)) \rightarrow (a \doteq m(r) \vee \neg s(0, 0))$$

Prädikatszeichen: $f(-, -)$, $u(-, -)$, $s(-, -)$

Funktionszeichen: $a, q, r, 0, g(-), p(-), m(-)$,

Häufig gestellte Frage

Bei den Ankreuzaufgaben:
Werden Punkte, Bruchteile von Punkten, abgezogen, wenn man eine Frage unbeantwortet läßt?

Antwort

Nein.

Wenn man in einer Zeile keine der angebotenen Antworten ankreuzt werden 0 Punkte berechnet.

Nur für angekreuzte falsche Antworten gibt es Punktabzug.

Gute Strategie:

Kreuzen Sie eine Antwort nur an, wenn Sie sich sicher sind, daß sie stimmt. Auf den Zufall zu hoffen, zahlt sich nicht aus.

ENDE