

Computational Logic

Algorithmische Logik

Prädikatenlogik

Ralf Moeller

Hamburg Univ. of Technology

Motivation

- Aussagenlogik: Aussagen als unteilbares Ganzes (atomare Formeln)
- Beispiel:
 - Wenn es regnet, werde ich naß: $R \rightarrow N$
 - Es regnet: R
 - Also: werde ich naß: N
- Wenn...dann: Wahrheitstabelle der Implikation
 - Wenn die Vorbedingung falsch ist, soll der ganze Implikationsterm wahr sein!
- Aussagenlogik recht mächtig (wir konnten z.B. schließen, daß Superman nicht existiert)

Probleme (1)

■ Gegeben

- Alle Metalle leiten den Strom (A)
- Kupfer ist ein Metall (B)

■ Gewünschte Folgerung

- Kupfer leitet den Strom (C)

■ In Formeln (vielleicht):

- $A \wedge B \rightarrow C$

Probleme (2)

- Gegeben
 - Alle Metalle leiten den Strom (A)
 - Eisen ist ein Metall (D)
- Gewünschte Folgerung
 - Eisen leitet den Strom (E)
- In Formeln (vielleicht):
 - $A \wedge D \rightarrow E$
- Neue Formeln für alle möglichen Metalle erforderlich!
- Gemeinsamkeiten nicht repräsentiert

Motivation (2)

- In Anwendungen ergeben sich Aussagen z.B. durch Analyse natürlichspracher Sätze (siehe Supermann-Beispiel)
- Sätze haben eine grammatische Struktur
- Wir betrachten einfach Satzstrukturen:
 - Subjekt-Prädikat: Ich schlafe
 - Subjekt-Prädikat-Objekt: Ich verfolge die Vorlesung

Subjekt-Prädikat-Strukturen

- Beispiel: Energie ist wertvoll
- Nutzung: Beschreibung von Eigenschaften
- Eigenschaften heißen Prädikate: Wertvoll-sein
- Wir betrachten einstellige Prädikate, die Eigenschaften eines „Subjekts“ (auch Individuum genannt) beschreiben
- Formalisierung:
 - Prädikate mit Großbuchstaben: P, Q, R, \dots
 - Subjekte mit Kleinbuchstaben: x, y, z, \dots

Einstellige Prädikate

- Notation: Wertvoll(x), P(x), ...
- Subjekte wie x bezeichnet man auch als Variablen
- Belegung der Variable: Übergang zur Aussage:
 - Wertvoll(blaue-Mauritius)
 - Die blaue Mauritius ist wertvoll
- Mögliche Belegungen für x aus vorgegebener Grundmenge

Subjekt-Prädikat-Objekt-Strukturen

- Beispiel: Edelgard ist mit Wolfgang verheiratet
- Variablen für Subjekt (Edelgard) und Objekt (Wolfgang) benötigt
- Prädikat beschreibt Beziehung (oder Relation) zwischen Subjekt und Objekt

Zweistellige Prädikate

- $\text{Verheiratet-mit}(x, y), P(x, y)$
- $\text{Verheiratet-mit}(\text{Edelgard}, \text{Wolfgang})$

Verallgemeinerung: n-stellige Prädikate

- Blankenese liegt zwischen Wedel und Altona.
- Prädikat: Zwischen
- Dreistellig
- Im allgemeinen: $P(x_1, x_2, \dots, x_n)$

Notation

- Manchmal ist die Notation $P(x,y)$ ungewohnt
- Beispiel: Prädikat Kleiner-als ($<$)
- Infix-Notation: $x < y$

Verknüpfung von Aussagen

- Beispiel: 2 und 3 sind Teiler von 6
- Steht für
 - 2 ist Teiler von 6
 - 3 ist Teiler von 6
- In Prädikatenschreibweise
 - $\text{Teiler-von}(2, 6) \wedge \text{Teiler-von}(3, 6)$
- Beispiel: 4 ist nicht Teiler von 6
 - $\neg \text{Teiler-von}(4, 6)$

Namen für Funktionen

- Bei vielen Anwendungen bequem:
Einführung von **Namen** für Funktionen
 - f, g, h, \dots (rein syntaktische Namen, Funktorvariablen)
 - f_1, f_2, f_3, \dots
- Funktoren haben eine festgelegte Stelligkeit und werden auf Variablen angewendet
- Terme: $f(x, y)$
 - Manchmal auch Infix-Schreibweise

Anwendung von Namen für Funktionen

- Beschreibung eines Objekts, das sich aus der Verknüpfung von x und y ergibt
- Beispiel:
 - Wenn y größer als 0 , dann ist $x+y$ größer als x
 - $(y > 0) \rightarrow ((x + y) > x)$
 - Oder: $\succ(y, 0) \rightarrow \succ(+ (x, y), x)$

Quantifizieren von Aussagen

- Eingangsbeispiel „Alle Metalle leiten den Strom“ nicht von Subjekt-Prädikat-Struktur erfaßt
- Einstelliges Prädikat Leitet-den-Strom bezieht sich nicht auf einen einzelnen Gegenstand
- Vielmehr wird ausgesagt, daß *alle* Subjekte, die Metalle sind, diese Eigenschaft besitzen
- Formalisierung durch **Allquantor** (auch: Universalquantor, Generalisator)
- Für *alle* x gilt: wenn x ein Metall ist, dann leitet x den Strom

Quantifizieren von Aussagen (2)

- Für alle x gilt: $\text{Metall}(x) \rightarrow \text{Leitet-Strom}(x)$
- Notation: $\forall x (\text{Metall}(x) \rightarrow \text{Leitet-Strom}(x))$
- Man beachte: Es muß nicht notwendigerweise überhaupt Metalle geben! Die Für-alle-Aussage ist ja dann nicht falsch
- Manchmal möchte man die **Existenz** aber fordern
- *Es gibt ein x* : $\text{Metall}(x)$, oder genauer aber synonym:
- *Es gibt mindestens ein x* : $\text{Metall}(x)$
- Notation: $\exists x (\text{Metall}(x))$ **Existenzquantor**

Syntax der Prädikatenlogik

Danksagung

- Die Folien zur Prädikatenlogik nach dem Buch "Logik für Informatiker" von Uwe Schöning wurden übernommen von **Javier Esparza** (<http://www.brauer.in.tum.de/lehre/logik/SS99/>)

Variablen, Symbole, Terme

Eine *Variable* hat die Form x_i mit $i = 1, 2, 3, \dots$

Ein *Prädikatsymbol* hat die Form P_i^k und ein *Funktionssymbol* hat die Form f_i^k mit $i = 1, 2, 3, \dots$ und $k = 0, 1, 2, \dots$. Hierbei heißt i jeweils der *Unterscheidungsindex* und k die *Stellenzahl* (oder Stelligkeit). Wir definieren nun die *Terme* durch einen induktiven Prozeß:

1. Jede Variable ist ein Term.
2. Falls f ein Funktionssymbol mit der Stellenzahl k , und falls t_1, \dots, t_k Terme sind, so ist auch $f(t_1, \dots, t_k)$ ein Term.

Hierbei sollen auch Funktionssymbole der Stellenzahl 0 eingeschlossen sein, und in diesem Fall sollen die Klammern wegfallen. Nullstellige Funktionssymbole heißen auch *Konstanten*.

Formeln

Nun können wir (wiederum induktiv) definieren, was *Formeln* (der Prädikatenlogik) sind.

1. Falls P ein Prädikatsymbol der Stelligkeit k ist, und falls t_1, \dots, t_k Terme sind, dann ist $P(t_1, \dots, t_k)$ eine Formel.
2. Für jede Formel F ist auch $\neg F$ eine Formel.
3. Für alle Formeln F und G sind auch $(F \wedge G)$ und $(F \vee G)$ Formeln.
4. Falls x eine Variable ist und F eine Formel, so sind auch $\exists xF$ und $\forall xF$ Formeln. Das Symbol \exists wird *Existenzquantor* und \forall *Allquantor* genannt.

Atomare Formeln nennen wir genau die, die gemäß 1. aufgebaut sind. Falls F eine Formel ist und F als Teil einer Formel G auftritt, so heißt F *Teilformel* von G .

Freie und gebundene Variablen, Aussagen

Alle Vorkommen von Variablen in einer Formel werden in *freie* und *gebundene* Vorkommen unterteilt. Dabei heißt ein Vorkommen der Variablen x in der Formel F gebunden, falls x in einer Teilformel von F der Form $\exists xG$ oder $\forall xG$ vorkommt. Andernfalls heißt dieses Vorkommen von x frei.

Eine Formel ohne Vorkommen einer freien Variablen heißt *geschlossen* oder eine *Aussage*.

Die *Matrix* einer Formel F ist diejenige Formel, die man aus F erhält, indem jedes Vorkommen von \exists bzw. \forall , samt der dahinterstehenden Variablen gestrichen wird. Symbolisch bezeichnen wir die Matrix der Formel F mit F^* .

Aufgabe

NF: Nicht-Formel F: Formel, aber nicht Aussage A: Aussage

	NF	F	A
$\forall xP(a)$			
$\forall x\exists y(Q(x,y) \vee R(x,y))$			
$\forall xQ(x,x) \rightarrow \exists xQ(x,y)$			
$\forall xP(x) \vee \forall xQ(x,x)$			
$\forall P(x)$			
$P(x) \rightarrow \exists x$			
$\forall\exists P(x)$			
$\forall x\neg\forall yQ(x,y) \wedge R(x,y)$			
$\exists z(Q(z,x) \vee R(y,z)) \rightarrow \exists y(R(x,y) \wedge Q(x,z))$			
$\exists x(\neg P(x) \vee P(a))$			
$P(x) \rightarrow \exists xP(x)$			
$\exists x\forall y((P(y) \rightarrow Q(x,y)) \vee \neg P(x))$			

Semantik der Prädikatenlogik

Struktur, passende Strukturen

Eine *Struktur* ist ein Paar $A = (U_A, I_A)$ wobei U_A eine beliebige aber **nicht leere** Menge ist, die die *Grundmenge* von A (oder der *Grundbereich*, der *Individuenbereich*, das *Universum*) genannt wird. Ferner ist I_A eine Abbildung, die

- jedem k -stelligen Prädikatensymbol P (das im Definitionsbereich von I_A liegt) ein k -stelliges Prädikat über U_A zuordnet,
- jedem k -stelligen Funktionssymbol f (das im Definitionsbereich von I_A liegt) eine k -stellige Funktion auf U_A zuordnet,
- jeder Variablen x (sofern I_A auf x definiert ist) ein Element der Grundmenge U_A zuordnet.

Sei F eine Formel und $A = (U_A, I_A)$ eine Struktur. A heißt zu F *passend*, falls I_A für alle in F vorkommenden Prädikatsymbole, Funktionssymbole und freien Variablen definiert ist.

Mit anderen Worten, der Definitionsbereich von I_A ist eine Teilmenge von $\{P_i^k, f_i^k, x_i \mid i = 1, 2, 3, \dots \text{ und } k = 0, 1, 2, \dots\}$, und der Wertebereich von I_A ist eine Teilmenge aller Prädikate und Funktionen auf U_A , sowie der Elemente von U_A . Wir schreiben abkürzend statt $I_A(P)$ einfach P^A , statt $I_A(f)$ einfach f^A und statt $I_A(x)$ einfach x^A .

Werte von Termen und Formeln in einer Struktur

Sei F eine Formel und A eine zu F passende Struktur. Für jeden Term t , den man aus den Bestandteilen von F bilden kann (also aus den Variablen und Funktionssymbolen), definieren wir nun den *Wert* von t in der Struktur A , den wir mit $A(t)$ bezeichnen. Die Definition ist wieder induktiv.

1. Falls t eine Variable ist (also $t = x$), so ist $A(t) = x^A$.
2. Falls t die Form hat $t = f(t_1, \dots, t_k)$ wobei t_1, \dots, t_k Terme und f ein k -stelliges Funktionssymbol ist, so ist
$$A(t) = f^A(A(t_1), \dots, A(t_k)).$$

Der Fall 2 schließt auch die Möglichkeit ein, daß f nullstellig ist, als t die Form hat $t = a$. In diesem Fall ist also $A(t) = a^A$.

Auf analoge Weise definieren wir (induktiv) den (*Wahrheits-*) *Wert* der Formeln F unter der Struktur A , wobei wir ebenfalls die Bezeichnung $A(F)$ verwenden.

- Falls F die Form hat $F = P(t_1, \dots, t_k)$ mit den Termen t_1, \dots, t_k und k -stelligem Prädikatsymbol P , so ist

$$A(F) = \begin{cases} 1, & \text{falls } (A(t_1), \dots, A(t_k)) \in P^A \\ 0, & \text{sonst} \end{cases}$$

- Falls F die Form $F = \neg G$ hat, so ist

$$A(F) = \begin{cases} 1, & \text{falls } A(G) = 0 \\ 0, & \text{sonst} \end{cases}$$

- Falls F die Form $F = (G \wedge H)$ hat, so ist

$$A(F) = \begin{cases} 1, & \text{falls } A(G) = 1 \text{ und } A(H) = 1 \\ 0, & \text{sonst} \end{cases}$$

- Falls F die Form $F = (G \vee H)$ hat, so ist

$$A(F) = \begin{cases} 1, & \text{falls } A(G) = 1 \text{ oder } A(H) = 1 \\ 0, & \text{sonst} \end{cases}$$

- Falls F die Form $F = \forall xG$ hat, so ist

$$A(F) = \begin{cases} 1, & \text{falls f\u00fcr alle } d \in U_A \text{ gilt : } A_{[x/d]}(G) = 1 \\ 0, & \text{sonst} \end{cases}$$

- Falls F die Form $F = \exists xG$ hat, so ist

$$A(F) = \begin{cases} 1, & \text{falls es ein } d \in U_A \text{ gibt mit : } A_{[x/d]}(G) = 1 \\ 0, & \text{sonst} \end{cases}$$

Hierbei bedeutet $A_{[x/d]}$ diejenige Struktur A' , die \u00fberall mit A identisch ist, bis auf die Definition von $x^{A'}$. Es sei n\u00e4mlich $x^{A'} = d$, wobei $d \in U_A = U_{A'}$ — unabh\u00e4ngig davon, ob I_A auf x definiert ist oder nicht.

Beispiel

Es sei P ein zweistelliges Prädikatensymbol, Q ein einstelliges Prädikatensymbol, a eine Konstante, f ein einstelliges Funktionssymbol, und es seien x, y, z Variablen. Eine Struktur, die diesen Symbolen Werte zuordnet, ist gegeben durch $A = (U_A, I_A)$ mit $U_A = \{7, 8, 9\}$. Die Interpretationsfunktion I_A ist wie folgt definiert:

$$a^A = 7$$

$$x^A = 8$$

$$y^A = 9$$

$$Q^A = \{7, 9\}$$

$$P^A = \{(7, 8), (7, 9), (8, 9)\}$$

$$f^A = \{(7, 8), (8, 9), (9, 7)\}$$

Beispiel (2)

■ Zu bestimmen ist der Wahrheitswert der folgenden Formeln bezüglich A :

| 1. $(Q(a) \wedge P(a, x)) \wedge (Q(f(a)) \wedge P(a, y))$

| 2. $P(a, y) \wedge \exists z P(z, a)$

| 3. $\exists z (Q(z) \wedge P(a, z))$

Modell, Gültigkeit, Erfüllbarkeit

Falls für eine Formel F und eine zu F passende Struktur A gilt $A(F) = 1$, so schreiben wir wieder $A \models F$.

Sprechweise: F *gilt* in A oder A ist *Modell* für F .

Falls jede zu F passende Struktur ein Modell für F ist, so schreiben wir $\models F$, andernfalls $\not\models F$.

Sprechweise: F ist (*allgemein-*)*gültig*.

Falls es mindestens ein Modell für die Formel F gibt, so heißt F *erfüllbar*, andernfalls *unerfüllbar*.

Aufgabe

G: Gültig E: Erfüllbar, aber nicht gültig U: Unerfüllbar

	G	E	U
$\forall xP(a)$			
$\exists x(\neg P(x) \vee P(a))$			
$P(a) \rightarrow \exists xP(x)$			
$P(x) \rightarrow \exists xP(x)$			
$\forall xP(x) \rightarrow \exists xP(x)$			
$\forall xP(x) \wedge \neg \forall yP(y)$			
$\forall x(P(x,x) \rightarrow \exists x \forall y P(x,y))$			
$\forall x \forall y (x = y \rightarrow f(x) = f(y))$			
$\forall x \forall y (f(x) = f(y) \rightarrow x = y)$			
$\exists x \exists y \exists z (f(x) = y \wedge f(x) = z \wedge y \neq z)$			

Noch einmal: Funktionen

- In der Semantik der Prädikatenlogik:
 - Spezifikation von I_A durch: $f^A = \{ (x, y), (y, z), \dots \}$
 - Schreibweise: $f(x) = y$ (x heißt Argument oder Parameter, y heißt Wert)
- Extensionale Definition problematisch da Tupelmenge potentiell unendlich
- Daher: Angabe der Tupelmenge durch Prädikatenlogische Formel (Beispiel)
 - $f^A = \{ (x, y) \mid y = x + 2 \}$
 - Schreibweise: $f(x) = x + 2$

Prädikate als Boole'sche Funktionen

■ Prädikatsnamen bezeichnen Mengen von Tupeln:

- Struktur (U_A, \cdot, I^A)
- Beispiel: $U_A = \{u, v, w\}$
- Beispiel: $P^A = \{(u, v), (v, w)\}$

■ Darstellung von Prädikaten als Funktionen:

- Beispiel: $P : U_A \times U_A \rightarrow B$ $B = \{0, 1\}$
- $P^A = \{(u, v, 1), (v, w, 1),$
 $(u, w, 0), (u, u, 0), (v, v, 0), (v, u, 0),$
 $(w, u, 0), (w, v, 0), (w, w, 0)\}$

Prädikatenlogik für „Arithmetik“

- Festlegung des Universums der Struktur
- Definition von Prädikaten mit Hilfe von „eingebauten“, vorausgesetzten Funktionen

Zahlen

N_0	auch: N	unsigned int , cardinal: natürliche Zahlen ab 0
N_1		natürliche Zahlen ab 1
Z		Intg, integer: ganze Zahlen
R		real, float: reelle Zahlen

Operationen auf natürlichen Zahlen

elementare
Prädikate

$$= : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{B}$$

$$\neq : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{B}$$

elementare
Operationen

$$+1 : \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$-1 : \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

Arithmetik

$$+ : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$- : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$\cdot : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$\text{div} : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$\text{mod} : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$+$, \cdot totale Funktionen

$-$, div , mod partielle Funktionen

Algebraische Axiome für $+$, $*$

- Einselemente
 - Kommutativität
 - Distributivität
 - Assoziativität
 -
-
- Axiome ermöglichen Umformulierungen von Termen

Weitere „eingebaute“ Prädikate

Vergleiche

$$\geq : \mathbf{N}_0 \times \mathbf{N}_0 \longrightarrow \mathbf{B}$$

$$> : \mathbf{N}_0 \times \mathbf{N}_0 \longrightarrow \mathbf{B}$$

$\geq, >$ totale Funktionen

Zusammenfassung, Kernpunkte



- Prädikatenlogik
 - Syntax, Formeln
 - Semantik, Belegung, Modell
 - Entscheidungsprobleme
 - Arithmetik in Prädikatenlogik

Folgerung und Äquivalenz

Eine Formel G heißt eine *Folgerung* der Formeln F_1, \dots, F_k falls für jede Struktur, die sowohl zu F_1, \dots, F_k als auch zu G passend ist, gilt:

Wenn A Modell von $\{F_1, \dots, F_k\}$ ist, dann ist A auch Modell von G .

Wir schreiben $F_1, \dots, F_k \models G$, falls G eine Folgerung von F_1, \dots, F_k ist.

Zwei Formeln F und G heißen (*semantisch*) *äquivalent*, falls für alle Strukturen A , die sowohl für F als auch für G passend sind, gilt $A(F) = A(G)$. Hierfür schreiben wir $F \equiv G$.

Aufgabe

1. $\exists y \forall x P(x, y)$

2. $\forall x \exists y P(x, y)$

	J	N
1. \models 2.		
2. \models 1.		

Äquivalenzen

Satz:

Seien F und G beliebige Formeln:

1. $\neg\forall xF \equiv \exists x\neg F$

$\neg\exists xF \equiv \forall x\neg F$

2. Falls x in G nicht frei vorkommt, gilt:

$(\forall xF \wedge G) \equiv \forall x(F \wedge G)$

$(\forall xF \vee G) \equiv \forall x(F \vee G)$

$(\exists xF \wedge G) \equiv \exists x(F \wedge G)$

$(\exists xF \vee G) \equiv \exists x(F \vee G)$

3. $(\forall xF \wedge \forall xG) \equiv \forall x(F \wedge G)$

$(\exists xF \vee \exists xG) \equiv \exists x(F \vee G)$

4. $\forall x\forall yF \equiv \forall y\forall xF$

$\exists x\exists yF \equiv \exists y\exists xF$

Aufgabe

	J	N
$\forall x \forall y F \equiv \forall y \forall x F$		
$\forall x \exists y F \equiv \exists x \forall y F$		
$\exists x \exists y F \equiv \exists y \exists x F$		
$\forall x F \vee \forall x G \equiv \forall x (F \vee G)$		
$\forall x F \wedge \forall x G \equiv \forall x (F \wedge G)$		
$\exists x F \vee \exists x G \equiv \exists x (F \vee G)$		
$\exists x F \wedge \exists x G \equiv \exists x (F \wedge G)$		

Beispiel

- Es seien *Animal*, *Vegetarian*, *Sheep*, *Cow*, und *Mad_cow* einstellige Prädikatensymbole. Die Symbole *EATS* und *PART_OF* seien zweistellige Prädikatensymbole, und es seien *x*, *y*, *z* Variablen. Wir betrachten folgende Menge *T* von Formeln:
- $T := \{ \neg x (\text{Sheep}(x) \wedge (\text{Animal}(x) \wedge \text{Vegetarian}(x))) , \neg x (\text{Cow}(x) \wedge (\text{Animal}(x) \wedge \text{Vegetarian}(x))) , \neg x (\text{Vegetarian}(x) \wedge (\neg \exists y (\text{EATS}(x,y) \wedge \text{Animal}(y)) \wedge \neg \exists y (\text{EATS}(x,y) \wedge \neg \exists z (\text{PART_OF}(y,z) \wedge \text{Animal}(z)))))) \}$

Beispiel (2)

- Annahme: Formeln seien Grundlage für die Spezifikation von Programmen für ein großes Softwareprojekt für das Landwirtschaftsministerium dar.
- Frage: Welche Kühe (eines bestimmten Bestandes, der als Parameter eingeht) sind beim Verzehr gesundheitsgefährdend?
- Hierzu wird folgende Formel verwendet:
$$f := \exists x (\text{Mad_cow}(x) \wedge (\text{Cow}(x) \wedge \exists y (\text{EATS}(x, y) \wedge \exists z \text{PART_OF}(y, z) \wedge \text{Sheep}(z))))$$

Beispiel (3)

- Zu erstellen: Programm, das die Menge M aller x berechnen soll, für die $\text{Mad_cow}(x)$ gilt
- Nach eingehender Beratung mit Ihren Mitarbeitern kommen Sie zu dem Schluß, daß Sie das Projekt, obwohl es lukrativ sein mag, aus Gewissensgründen nicht annehmen werden, da die Menge M immer leer sein wird.
- Sie zeigen, daß die Formel $\exists x \text{ Mad_cow}(x)$ unerfüllbar bzgl. $T \cup \{f\}$ ist

Prädikatenlogik für „Arithmetik“

- Festlegung des Universums der Struktur
- Definition von Prädikaten mit Hilfe von „eingebauten“, vorausgesetzten Funktionen

Zahlen

N_0 auch: N unsigned int , cardinal: natürliche Zahlen ab 0

N_1 natürliche Zahlen ab 1

Z Intg, integer: ganze Zahlen

R real, float: reelle Zahlen

Operationen

auf natürlichen Zahlen

elementare Prädikate

$$= : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{B}$$

$$\neq : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{B}$$

elementare Operationen

$$+1 : \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$-1 : \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

Arithmetik

$$+ : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$- : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$\cdot : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$\text{div} : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$$\text{mod} : \mathbb{N}_0 \times \mathbb{N}_0 \longrightarrow \mathbb{N}_0$$

$+$, \cdot totale Funktionen

$-$, div , mod partielle Funktionen

Algebraische Axiome für $+$, $*$

- Einselemente
 - Kommutativität
 - Distributivität
 - Assoziativität
 -
-
- Axiome ermöglichen Umformulierungen von Termen

Weitere „eingebaute“ Prädikate

Vergleiche

$$\geq : \mathbf{N}_0 \times \mathbf{N}_0 \longrightarrow \mathbf{B}$$

$$> : \mathbf{N}_0 \times \mathbf{N}_0 \longrightarrow \mathbf{B}$$

$\geq, >$ totale Funktionen

Wie hängen die Funktionen und Prädikate zusammen?

- Festlegung der Eigenschaften durch sog. Axiome
- Aus den durch Axiome festgelegten Eigenschaften sind Folgerungen (\models) möglich
- Menge von Axiomen heißt auch "Theorie"
- Üblich: Notation der Axiome als Formeln
- Zusammenhang zwischen " \rightarrow " und " \models "

Ungleichungen

Rechnen mit Ungleichungen

i, j, k seien ganze Zahlen

Axiome:

transitiv

$$i \leq j \wedge j \leq k \rightarrow i \leq k$$

reflexiv

$$i \leq i$$

antisymmetrisch

$$i \leq j \wedge j \leq i \rightarrow i = j$$

$$i \leq j \wedge 0 \leq k \rightarrow i * k \leq j * k$$

transitiv

$$i < j \wedge j < k \rightarrow i < k$$

Weitere Axiome:

$$\neg(i < i)$$

$$\neg(i < j \wedge j < i)$$

$$i < j \rightarrow i + k < j + k$$

$$i < j \wedge 0 \leq k \rightarrow i * k \leq j * k$$

$$i < j \wedge 0 < k \rightarrow i * k < j * k$$

$$i < j \rightarrow i + 1 \leq j$$

Noch einmal: Natürliche Zahlen

- Wir betrachten Strukturen $A = (U_A, I_A)$
- Ausgezeichnete Menge N (auch N_0 genannt) der natürlichen Zahlen einschließlich 0, $N \subseteq U_A$
- Annahme: Prädikat N zum Test, ob ein Objekt $n \in U_A$ aus N ist

Vereinbarung von abkürzenden Schreibweisen

- $\forall n \in \mathbb{N} . P(n)$ für $\forall n (\mathbb{N}(n) \Rightarrow P(n))$ (dto. für \mathbb{Z})
- $\forall a \in \mathbb{N} \in \mathbb{N} . P(n)$ für $\forall n (((\mathbb{N}(n) \Rightarrow (a \in \mathbb{N})) \Rightarrow (n \in \mathbb{N})) \Rightarrow P(n))$
- $\forall x \in M . F(x)$

- $\exists n \in \mathbb{N} . P(n)$ für $\exists n (\mathbb{N}(n) \wedge P(n))$ (dto. für \mathbb{Z})
- $\exists a \in \mathbb{N} \in \mathbb{N} . P(n)$ für $\exists n (((\mathbb{N}(n) \wedge (a \in \mathbb{N})) \wedge (n \in \mathbb{N})) \wedge P(n))$
- $\exists x \in M . F(x)$

Beispiel

Aussage

„alle Aldi-PCs sind schlecht gebaut“

Grundmenge

PCs

**elementare
Prädikate**

$vonAldi(x)$

$gut(x)$

Formel

mit All-Quantor

$\forall pc \in PCs \bullet vonAldi(pc) \Rightarrow \neg gut(pc)$

Aussage	„es gibt eine gerade Primzahl“
Grundmenge	\mathbb{N}_0
elementare Prädikate	$gerade(x)$ $prim(x)$
Formel	mit Existenz-Quantor $\exists n \in \mathbb{N}_0 \bullet gerade(n) \wedge prim(n)$

Tautologien/Gesetze (1)

$$(1) \quad \forall i \in \{\} \bullet P(i) \leftrightarrow \text{true}$$

$$(2) \quad \exists i \in \{\} \bullet P(i) \leftrightarrow \text{false}$$

$$(3) \quad \exists i \in M \bullet \neg P(i) \leftrightarrow \neg \forall i \in M \bullet P(i)$$

$$\forall i \in M \bullet \neg P(i) \leftrightarrow \neg \exists i \in M \bullet P(i)$$

Verallgemeinerung von de Morgan

$$(4) \quad \forall i \in M \bullet P(i) \wedge j \in M \rightarrow P(j)$$

$$(5) \quad j \in M \wedge P(j) \rightarrow \exists i \in M \bullet P(i)$$

Tautologien/Gesetze (2)

$$(1a) \quad \forall a \leq i < a \bullet P(i) \Leftrightarrow \text{true}$$

$$(2a) \quad \exists a \leq i < a \bullet P(i) \Leftrightarrow \text{false}$$

$$(3a) \quad \exists a \leq i < b \bullet \neg P(i) \Leftrightarrow \neg \forall a \leq i < b \bullet P(i)$$

$$\forall a \leq i < b \bullet \neg P(i) \Leftrightarrow \neg \exists a \leq i < b \bullet P(i)$$

$$(4a) \quad \forall a \leq i < b \bullet P(i) \wedge a \leq j < b \rightarrow P(j)$$

$$(5a) \quad a \leq j < b \wedge P(j) \rightarrow \exists a \leq i < b \bullet P(i)$$

$$(6a) \quad \forall a \leq i < b \bullet P(i) \wedge \forall b < i < c \bullet P(i) \\ \Leftrightarrow \forall a \leq i < c \bullet P(i)$$

Grundmenge $i \in S$

Intervall häufiger Sonderfall von Mengen:
Intervall aus den ganzen Zahlen

Notation $a \leq i < b$

alternative

Notation für Formeln mit Quantoren

$\forall a \leq i < b \bullet P(i)$ halb offenes Intervall

$\forall a \leq i \leq b \bullet P(i)$ abgeschlossenes Intervall

$\forall a \leq i \bullet P(i)$ unbeschränktes Intervall

Felder $f : \text{array } [0 .. n - 1] \text{ of } \textit{Element}$

f ist eine Variable zum Speichern einer ganzen Menge von Werten

Referenzierung $f[i], f(i), f_i$ Zugriff auf i -te Element in f

Indexbereich ein Intervall (hier $0 \leq i < n$)

Beispiele	für Prädikate mit \forall -Quantoren
gegeben	zwei Felder a und b vom Typ array $[0 .. n-1]$ of Z
Aufgabe	Wie sehen die zugehörigen Prädikate aus?
(1)	a ist eine exakte Kopie von b , alle Elemente sind gleich
(2)	Für alle i ist das i -te Element von a kleiner als das i -te Element von b
(3)	Jedes Element von a ist kleiner als jedes Element von b

- (4) Wenn die Elemente von a in aufsteigender Reihenfolge sortiert sind, so auch die Elemente von b
- (5) Alle Elemente von a sind untereinander verschieden
- (6) Jedes Element von a ist von jedem Element von b verschieden

Beispiele

für Prädikate mit \exists - und \forall -Quantoren

gegeben

zwei Felder a und b

vom Typ array $[0 .. n-1]$ of Z

Aufgabe

Wie sehen die zugehörigen Prädikate aus?

(1)

Einige Elemente von a sind ungleich 0

(2)

das Feld a ist nicht in aufsteigender Reihenfolge sortiert

(3)

mindestens ein Element von a ist größer als alle Elemente von b

(4)

Jedes Element von b ist eine Kopie eines Elements von a

- (5) b enthält alle Zahlen von 0 bis $n-1$ (eine Permutation aller Zahlen von 0 bis $n-1$)
- (6) b zeigt die numerische Ordnung von a an, d.h. $b[0]$ ist der Index auf das kleinste Element von a , $b[1]$ der Index auf das zweitkleinste Element usw.
- (7) Wenn alle Elemente in a paarweise verschieden sind, dann enthält b die Elemente von a in sortierter Reihenfolge
- (8) b enthält die Elemente von a in sortierter Reihenfolge

Negation	von Prädikaten mit Quantoren
Aussage	<i>„alle Aldi-PCs sind schlecht gebaut“</i>
Frage	Welche der folgenden Aussagen ist die Negation dieser Aussage
(1)	Alle nicht-Aldi-PCs sind gut gebaut
(2)	Alle Aldi-PCs sind gut gebaut
(3)	Einige Aldi-PCs sind gut gebaut
(4)	Einige Aldi-PCs sind schlecht gebaut
(5)	Einige nicht-Aldi-PCs sind schlecht gebaut
(6)	Einige nicht-Aldi-PCs sind gut gebaut

Algorithmen vs. Funktionen

- Statt extensionaler Spezifikation einer Funktion
- Spezifikation mit Hilfe einer Formel bzw. eines Prädikates (intensionale Spezifikation)
- Algorithmen berechnen Funktionen
- Berechnungsvorschrift (ggf. für jeden möglichen Parameter) erforderlich (-> Algorithmus)
- Berechnungsvorschrift u.U. nicht einfach zu finden (-> Korrektheit, Vollständigkeit, Terminierung)
- Systematische Entwicklung notwendig

Algorithmen

- Verwendung einer beliebigen Menge von "Variablen" $\{x, y, z, \dots\}$
- Elementare Anweisungen
 - Zuweisung (Bsp. " $x := 2$ ")
 - Fallunterscheidung
 - Schleife
- Algorithmus: Sequenz von elementaren Anweisungen
- Idee: Nach Ausführung des Algorithmus steht Funktionswert in festgelegter Variable

Zusammenfassung, Kernpunkte



■ Prädikatenlogik

- Syntax, Formeln
- Semantik, Belegung, Modell
- Entscheidungsprobleme
- Äquivalente Transformation von Formeln

■ Anwendungsmotivation:

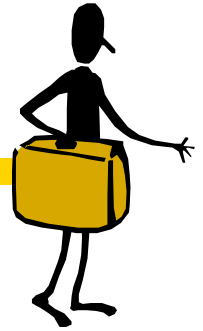
- Konzeptuelle Datenmodellierung

Was kommt beim nächsten Mal?



- Elemente von Programmiersprachen:
 - Kontrollstrukturen
- Verifikation von Programmen

Zusammenfassung, Kernpunkte



- Prädikatenlogik
- Folgerbarkeit und Äquivalenz
- Äquivalente Transformation von Formeln
- Grundlagen der logischen Modellierung