Grundlagen der Informatik

1. Semester, 1996

1.1. Situationen durch Formeln beschreiben

1.1.1. Situationen aussagenlogisch beschreiben

1.1.2. Situationen prädikatenlogisch beschreiben

1.1.3. Situationen quantorenlogisch beschreiben

Beispiel: Logelei (1)

"Wer von Euch hat den Ball in mein Fenster geworfen? schreit der Mann voller Zorn. Zitternd stehen die vier Kinder da.

Arno sagt: "Emil war es."

Emil sagt: "Gustav hat es getan."

Fritz sagt: "Ich war es nicht."

Gustav sagt: "Emil lügt."

Ein Passant, der den Wurf beobachtet hat, sagt: "Eins der Kinder war es, aber Vorsicht: Nur eins der Kinder sagt die Wahrheit."

Wer hat den Ball geworfen??

Beispiel: Logelei (2)

Arno sagt: "Emil war es."

Emil sagt: "Gustav hat es getan."

Fritz sagt: "Ich war es nicht."

Gustav sagt: "Emil lügt."

Lösung: betrachte nacheinander die Fälle:

1. Arno sagt die Wahrheit. Dann lügen Emil, Fritz und Gustav. Wenn aber Gustav lügt, wenn er sagt "Emil lügt", dann sagt Emil die Wahrheit. Das ist ein Widerspruch.
2. Emil sagt die Wahrheit. Dann lügen Arno, Fritz und Gustav. Wenn aber Fritz lügt, wenn er sagt "Ich war es nicht", dann lügt Emil. Das ist ein Widerspruch.
3. Fritz sagt die Wahrheit. Dann lügen Arno, Emil, und Gustav. Wenn aber Gustav lügt, wenn er sagt "Emil lügt", dann sagt Emil die Wahrheit. Das ist ein Widerspruch.

Folgerung:

Gustav sagt die Wahrheit. Also lügt Fritz. Also hat Fritz den Ball geworfen.

Beispiel: Logelei (3)

- Abkürzungen für Aussagen:

  Wa(a), Wa(e), Wa(f), Wa(g) für "Arno, (Emil, Fritz, Gustav) sagt die Wahrheit"

  Tä(a), Tä(e), Tä(f), Tä(g) für "Arno, (Emil, Fritz, Gustav) war der Täter"

  Beispiel: Arno sagt: Tä(e)

- Formalisierung von Aussagen

`Arno sagt:"Emil war es"`	Wa(a) genau dann wenn Tä(e)
`Emil sagt:"Gustav hat es getan"`	Wa(e) genau dann wenn Tä(g)
`Fritz sagt:"Ich war es nicht"`	Wa(f) g. d. w. nicht Tä(f)
`Gustav sagt:"Emil lügt"`	Wa(g) g. d. w. nicht Wa(e)

Beispiel: Logelei (4)

- Formalisierung von Aussagen

`"Nur eins der Kinder sagt die Wahrheit"`

Wenn Wa(a) dann nicht Wa(e) und nicht Wa(f) und nicht Wa(g)

Wenn Wa(e) dann nicht Wa(a) und nicht Wa(f) und nicht Wa(g)

Wenn Wa(f) dann nicht Wa(a) und nicht Wa(e) und nicht Wa(g)

Wenn Wa(g) dann nicht Wa(a) und nicht Wa(e) und nicht Wa(f)

`"Nur eins der Kinder ist der Täter"`

Wenn Tä(a) dann nicht Tä(e) und nicht Tä(f) und nicht Tä(g)

Wenn Tä(e) dann nicht Tä(a) und nicht Tä(f) und nicht Tä(g)

Wenn Tä(f) dann nicht Tä(a) und nicht Tä(e) und nicht Tä(g)

Wenn Tä(g) dann nicht Tä(a) und nicht Tä(e) und nicht Tä(f)

Aussagen

Definition 1.1.

• Eine Aussage ist eine Form, die wahr oder falsch sein kann, aber nicht beides.

• Aussagen werden mit Aussagesymbolen bezeichnet, z.B. P,Q, Wa(e), Tä(e),…

• Die Wahrheitswerte wahr (Abk.: W) und falsch (Abk.: F) sind auch Aussagen.

• Aussagen werden durch Wahrheitsfunktionen (Boolsche Funktionen) verknüpft.

• Eine Wahrheitsfunktion ist eine Funktion mit W und F als Argumenten.

• Im folgenden werden die 5 Wahrheitsfunktionen

nicht, und, oder, wenn dann, genau dann wenn,

benutzt und über Wahrheitstafeln definiert.

• Verknüpfungen und ihre Zeichen heißen auch (aussagenlogische) Junktoren.

Wahrheitsfunktionen (1)

Verknüpfung	**Zeichen**	**Wahrheitstafel**	**sonst übliche Symbole**
nicht	$\neg$	$\begin{array}{c\|cc} \neg & \mathbf{W} & \mathbf{F} \\ \hline & F & W \end{array}$	$\sim$, $-$
und	$\wedge$	$\begin{array}{c\|cc} \wedge & \mathbf{W} & \mathbf{F} \\ \hline W & W & F \\ F & F & F \end{array}$	&, •, aneinanderschreiben
oder	$\vee$	$\begin{array}{c\|cc} \vee & \mathbf{W} & \mathbf{F} \\ \hline W & W & W \\ F & W & F \end{array}$	+
wenn dann	$\rightarrow$	$\begin{array}{c\|cc} \rightarrow & \mathbf{W} & \mathbf{F} \\ \hline W & W & F \\ F & W & W \end{array}$	$\Rightarrow$, $\supset$
genau dann wenn	$\leftrightarrow$	$\begin{array}{c\|cc} \leftrightarrow & \mathbf{W} & \mathbf{F} \\ \hline W & W & F \\ F & F & W \end{array}$	$\equiv$, $\Leftrightarrow$, $\sim$

Wahrheitsfunktionen (2)

$\neg p$	$= W$	genau dann wenn	nicht $p = W$	
$p \wedge q$	$= W$	genau dann wenn	$p = W$	und $q = W$
$p \vee q$	$= W$	genau dann wenn	$p = W$	oder $q = W$
$p \rightarrow q$	$= W$	genau dann wenn	wenn $p = W$	dann $q = W$
$p \leftrightarrow q$	$= W$	genau dann wenn	$p = W$	gdw $q = W$

Aussagenlogische Formeln

Definition 1.2.

Eine aussagenlogische Formel wird definiert durch:

(1) Die Wahrheitswerte und Aussagen sind Formeln.

(2) Sind A und B Formeln, so auch

$\neg A$	(die Negation von A: "nicht A")
$(A \wedge B)$	(die Konjunktion von A und B: "A und B")
$(A \vee B)$	(die Disjunktion von A und B: "A oder B")
$(A \rightarrow B)$	(die Implikation (Konditional) von A und B: "wenn A dann B")
$(A \leftrightarrow B)$	(das Bikonditional von A und B: "A genau dann wenn B")

Beispiele

Formeln: $(P \wedge \neg P)$, $((P \rightarrow F) \leftrightarrow \neg P)$, $(((P \wedge \neg Q) \rightarrow R) \leftrightarrow (P \rightarrow (Q \rightarrow R)))$

keine Formeln: $(\wedge P \neg P)$, $((P \rightarrow F) \neg \leftrightarrow P)$, $(((P \wedge \neg Q) \rightarrow R) \leftrightarrow (P \rightarrow (Q \rightarrow R))$

Klammerkonventionen

- $\neg$ bindet stärker als $\wedge$, $\vee$ $\rightarrow$, $\leftrightarrow$
- $\wedge$ und $\vee$ binden stärker als $\rightarrow$, $\leftrightarrow$
- $\wedge$ und $\vee$ binden gleich stark
- $\rightarrow$ und $\leftrightarrow$ binden gleich stark

Beispiele

$P \wedge \neg P$	statt $(P \wedge (\neg P))$,
$(P \rightarrow F) \leftrightarrow \neg P$	statt $((P \rightarrow F) \leftrightarrow (\neg P))$
$(P \wedge \neg Q \rightarrow R) \leftrightarrow (P \rightarrow (Q \rightarrow R))$	statt $(((P \wedge (\neg Q)) \rightarrow R) \leftrightarrow (P \rightarrow (Q \rightarrow R)))$

Formalisieren: Beispiel Logelei

B1	$Wa(a) \leftrightarrow Tä(e)$	`Arno sagt:"Emil war es"`
B2	$Wa(e) \leftrightarrow Tä(g)$	`Emil sagt:"Gustav hat es getan"`
B3	$Wa(f) \leftrightarrow \neg Tä(f)$	`Fritz sagt:"Ich war es nicht"`
B4	$Wa(g) \leftrightarrow \neg Wa(e)$	`Gustav sagt:"Emil lügt"`
B5	$Wa(a) \rightarrow \neg Wa(e) \wedge \neg Wa(f) \wedge \neg Wa(g)$	
B6	$Wa(e) \rightarrow \neg Wa(a) \wedge \neg Wa(f) \wedge \neg Wa(g)$	
B7	$Wa(f) \rightarrow \neg Wa(a) \wedge \neg Wa(e) \wedge \neg Wa(g)$	
B8	$Wa(g) \rightarrow \neg Wa(a) \wedge \neg Wa(e) \wedge \neg Wa(f)$	
B9	$Wa\ (a) \vee Wa\ (e) \vee Wa(f) \vee Wa\ (g)$	
B10	$Tä(a) \leftrightarrow \neg Tä(e) \wedge \neg Tä(f) \wedge \neg Tä(g)$	
B11	$Tä(e) \leftrightarrow \neg Tä(a) \wedge \neg Tä(f) \wedge \neg Tä(g)$	
B12	$Tä(f) \leftrightarrow \neg Tä(a) \wedge \neg Tä(e) \wedge \neg Tä(g)$	
B13	$Tä(g) \leftrightarrow \neg Tä(a) \wedge \neg Tä(e) \wedge \neg Tä(f)$	

Belegung & Wahrheitswerte

- Wahrheitswert einer aussagenlogischen Formel hängt vom Wahrheitswert ihrer Teilformeln ab

  Bsp.: Tä(e) wird mit W belegt,

  Tä(a) wird mit F belegt,

  dann hat die Formel Tä(a) $\wedge$ Tä(e) den Wahrheitswert F

- Belegung der aussagenlogischen Formel ordnet einen Wahrheitswert zu

- Syntax von Formeln:

  Regeln, nach denen Formeln aufgebaut werden

  Bsp.: Definition 1.2.

- Semantik von Formeln:

  Regeln, nach denen Formeln ausgewertet werden

  Bsp.: Definition 1.3.

Belegung (1)

Definition 1.3.

- Eine Belegung (einer Formel A) ist eine Abbildung

  β: Menge der Aussagensymbole (von A) $\rightarrow$ {W, F}

  die jedem Aussagensymbol P (in A) eine Wahrheitswert β(P) zuordnet
- Der (Wahrheits-)Wert von Formeln unter der Belegung β wird mit Hilfe von Auswertungsregeln definiert , wobei ~> für 'ersetze' steht
  (1) Jedes Aussagensymbol wird durch seinen Wahrheitswert unter β ersetzt

  P ~> β(P)

  (2) Verknüpfungszeichen mit Wahrheitswerten als Argumenten werden gemäß Definition 1.1. ausgewertet: ¬ W ~> F, ¬ F ~> W, W $\wedge$ W ~> W, F $\wedge$ W ~> F, …
- Der Wahrheitswert, der sich aus einer Formel A bei der Auswertung unter der Belegung β ergibt, heißt Wert von A unter β ; Schreibweise: $\text{Wert}_\beta(A)$
- Die Formel A ist wahr (bzw. falsch) unter β, wenn $\text{Wert}_\beta(A)$ =W (bzw. =F) ist.
- Eine Formelmenge X ist wahr unter β, wenn jede Formel aus X wahr ist unter β.

Belegung (2)

jetzt:

$\neg P$	gdw	nicht P	
$P \wedge Q$	gdw	P	und Q
$P \vee Q$	gdw	P	oder Q
$P \rightarrow Q$	gdw	wenn P	dann Q
$P \leftrightarrow Q$	gdw	P	gdw Q

statt:

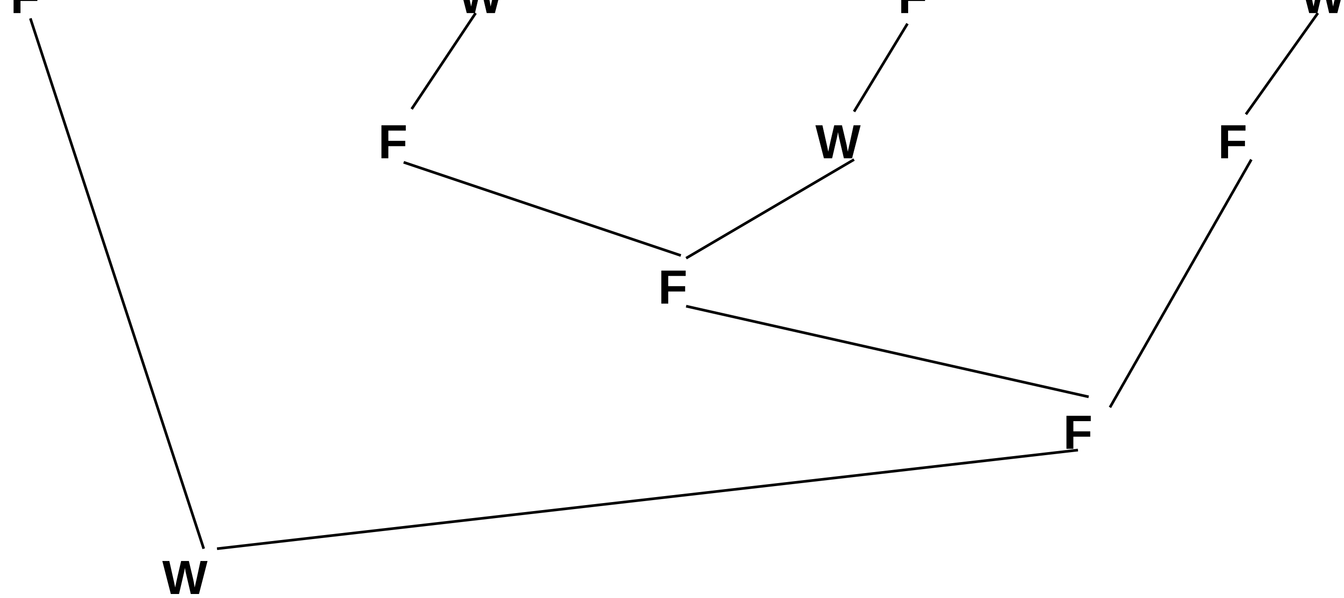
$\neg p$	= W	gdw	nicht p = W	
$p \wedge q$	= W	gdw	p = W	und q = W
$p \vee q$	= W	gdw	p = W	oder q = W
$p \rightarrow q$	= W	gdw	wenn p = W	dann q = W
$p \leftrightarrow q$	= W	gdw	p = W	gdw q = W

Bestimmung von Wahrheitswerten (1)

1. Baummethode:

Tä(a) ↔ ¬ Tä(e) ∧ ¬ Tä(f) ∧ ¬ Tä(g)

Belegung: **F** **W** **F** **W**



⇒ Tä(a) ↔ ¬ Tä(e) ∧ ¬ Tä(f) ∧ ¬ Tä(g) gilt für die Belegung

"Gustav und Emil sind Täter, Arno und Fritz sind unschuldig"

Bestimmung von Wahrheitswerten (2)

2. Zeilenmethode:

P Q	(P	→	Q)	∧	(Q	→	P)
W **W**	**W**	**W**	**W**	**W**	**W**	**W**	**W**
W **F**	**W**	**F**	**F**	**F**	**F**	**W**	**W**
F **W**	**F**	**W**	**W**	**F**	**W**	**F**	**F**
F **F**	**F**	**W**	**F**	**W**	**F**	**W**	**F**

Formalisieren: Inspektor Craig (1)

Ein Fall aus den Akten von Inspektor Craig: (aus R. Smullyan: "Wie heißt dieses Buch?")

"Was fängst du mit diesen Fakten an?" fragt Inspektor Craig den Sergeant McPherson.

(1) Wenn A schuldig und B unschuldig ist, so ist C schuldig.

(2) C arbeitet niemals allein.

(3) A arbeitet niemals mit C

(4) Niemand außer A, B oder C war beteiligt, und mindestens einer von ihnen ist schuldig

Der Sergeant kratzte sich den Kopf und sagte: "Nicht viel, tut mir leid, Sir. Können Sie nicht aus diesen Fakten schließen, wer unschuldig und wer schuldig ist? " "Nein", entgegnete Craig, "aber das Material reich aus, um wenigstens einen von ihnen zu beschuldigen."

Wer ist auf jeden Fall schuldig?

Formalisieren: Inspektor Craig (2)

Mr. McGregor, ein Londoner Ladeninhaber, rief bei Scotland Yard an und teilte mit, daß sein Laden ausgeraubt worden sei. Drei Verdächtige, A, B und C, wurden zum Verhör geholt. Folgende Tatbestände wurden ermittelt:

(1) Jeder der Männer A, B und C war am Tag des Geschehens in dem Laden gewesen, und kein anderer hatte den Laden an dem Tag betreten.

(2) Wenn A schuldig ist, so hat er genau einen Komplizen.

(3) Wenn B unschuldig ist, so ist auch C unschuldig.

(4) Wenn genau zwei schuldig sind, dann ist A einer von ihnen.

(5) Wenn C unschuldig ist, so ist auch B unschuldig.

Wen hat Inspektor Craig beschuldigt?

(aus R. Smullyan: "Wie heißt dieses Buch?")

1.1. Situationen durch Formeln beschreiben

- 1.1.1. Situationen aussagenlogisch beschreiben
- 1.1.2. Situationen prädikatenlogisch beschreiben
- 1.1.3. Situationen quantorenlogisch beschreiben

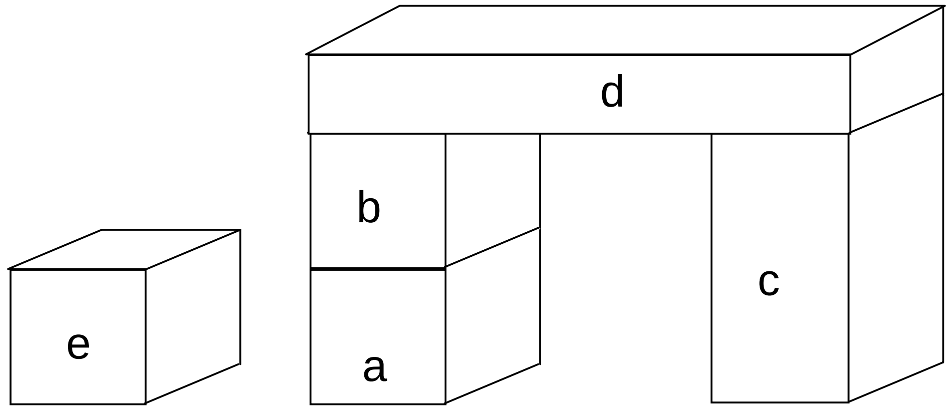
offene (quantorenfreie) Prädikatenlogik

- Elemente der Bereiche sind Daten
- meist gibt es verschiedene Sorten von Daten, z.B. bool, integer, Bäume, …
- Aussagen über Operationen auf Daten werden durch Prädikate beschrieben
- Prädikate treffen auf Daten zu oder nicht
- ein so strukturierter Bereich wird Struktur genannt
- die Menge der verwendeten Symbole ist die Signatur der Sprache/Struktur
- aus Daten-und Operationssymbolen werden Terme zusammengesetzt
- Variablen werden als Platzhalter für Datenmengen benutzt
- atomare Formeln sind Prädikatsymbole mit Termen als Argumenten
- Formeln sind durch Verknüpfungszeichen zusammengesetzte atomare Formeln
- eine Formel ist gültig in einer Struktur, wenn sie für alle Daten, die wir für die Variablen einsetzen, wahr ist

Beispiel: Blockwelt (1)

Problem:

Programm soll in einer Entwurfszeichnung Blöcke, Pfeiler und Bögen identifizieren



Beispiel: in der Zeichnung gilt:

- a,b,c,d,e sind Blöcke
- jeder Block, der auf dem Boden steht, ist ein Pfeiler
- a,b bildet einen Pfeiler, ebenso a,b,d und c,d
- sonst gibt es keine Pfeiler
- d auf a,b und auf c bilden einen Bogen

Beispiel: Blockwelt (2)

Prinzipien**, nach denen Bögen und Pfeiler aufgebaut sind**

C1 Wenn ein Block auf zwei Pfeilern liegt, so bilden sie zusammen einen Bogen.

C2 Jeder Block, der auf dem Boden steht, ist ein Pfeiler.

C3 Ein Block, der auf einem Pfeiler liegt, bildet mit ihm wieder einen Pfeiler.

C4 Ein Block, der auf dem obersten Block eines Pfeilers liegt, liegt damit auch auf dem Pfeiler.

genauere **Formulierung**:

C1 Wenn x ein Block, u ein Pfeiler und v ein Pfeiler ist und wenn x auf v und auf u liegt, dann ist das Gebilde aus x und v und u ein Bogen.

C2 Wenn x ein Block ist, der auf dem Boden steht, so ist das Gebilde aus x allein ein Pfeiler.

C3 Wenn x ein Block ist, der auf einem Pfeiler u liegt, dann bilden x und u einen Pfeiler.

C4 Wenn x auf y liegt und y der oberste Block eines Pfeilers ist, dann liegt x auch auf dem Pfeiler.

Beispiel: Blockwelt (3)

Symbole **für die Operationen und Prädikate**

• Block(x)	x ist ein Block
• Pfeiler(x)	x ist ein Pfeiler
• Bogen(u)	u ist ein Bogen
• auf-Boden(x)	x steht auf dem Boden
• liegt-auf(x,y)	x liegt auf y
• aus-einem(x)	das Gebilde aus x allein
• aus-zwei(x,y)	das Gebilde aus x auf y
• aus-drei(x,u,v)	das Gebilde aus x auf u und v

Beispiel: Blockwelt (4)

Formalisierung **von C1-C4**:

C1 **Block(x)** $\wedge$ **Pfeiler(u)** $\wedge$ **Pfeiler(v)** $\wedge$ **liegt-auf(x,v)** $\wedge$ **liegt-auf(x,u)**

$\rightarrow$ **Bogen(aus-drei(x,v,u))**

Wenn x ein Block, u ein Pfeiler und v ein Pfeiler ist und wenn x auf v und auf u liegt, dann ist das Gebilde aus x und v und u ein Bogen.

C2 **Block(x)** $\wedge$ **auf-Boden(x)** $\rightarrow$ **Pfeiler(aus-einem(x))**

Wenn x ein Block ist, der auf dem Boden steht, so ist das Gebilde aus x allein ein Pfeiler.

C3 **Block(x)** $\wedge$ **Pfeiler(u)** $\wedge$ **liegt-auf(x,u)** $\rightarrow$ **Pfeiler(aus-zwei(x,u))**

Wenn x ein Block ist, der auf einem Pfeiler u liegt, dann bilden x und u einen Pfeiler.

C4 **Block(x)** $\wedge$ **Block(y)** $\wedge$ **Pfeiler(aus-zwei(y,u))**

$\wedge$ **liegt-auf(x,aus-einem(y))** $\wedge$ **liegt-auf(y,u)**

$\rightarrow$ **liegt-auf(x,aus-zwei(y,u))**

Wenn x auf y liegt und y der oberste Block eines Pfeilers ist, dann liegt x auch auf dem Pfeiler.

Beispiel: Blockwelt (5)

Formalisierung **des Bildes**:

C5 Block(a)

C6 Block(b)

C7 Block(c)

C8 Block(d)

C9 Block(e)

C10 liegt-auf(b,aus-einem(a))

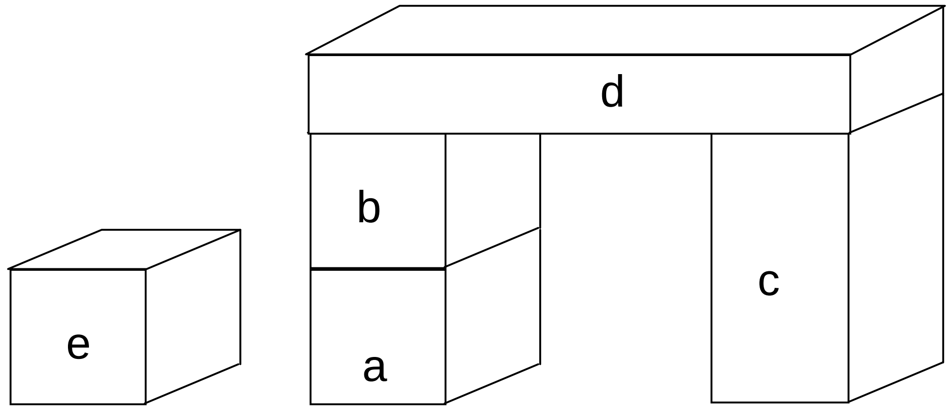
C11 liegt-auf(d,aus-einem(b))

C12 liegt-auf(d,aus-einem(c))

C13 auf-Boden(a)

C14 auf-Boden(c)

C15 auf-Boden(e)



Folgerung aus C1-C15:

Bogen(aus-drei(d,aus-zwei(b,aus-einem(a)),aus-einem(c)))

Struktur (1)

Definition 1.4.

- Eine Struktur *M* ist durch Daten verschiedener Sorten sowie durch Funktionen und Prädikate darauf gegeben.
- Die Daten der Sorte s bilden den (nichtleeren!) Bereich Ms der Sorte s.
- Alle Daten zusammen bilden den Bereich M von *M*.
- Eine Operation auf M ist eine Funktion

  f: Ms1 x…x Msn -> Ms,

  wobei s1,…,sn und s Sorten sind.
- Der Argumentbereich von f ist die Menge Ms1 x … x Msn
- Der Wertebereich von f ist die Menge Ms
- Das Tupel (s1,…,sn) ist der Argumenttyp von f.
- Das Sortensymbol s ist der Werttyp von f.
- Die Zahl n > 0 ist die Stelligkeit von f.

Struktur (2)

Definition 1.4.

- Eine Prädikat auf M ist eine Funktion

  P: Ms1 x…x Msn -> {W,F},

  wobei s1,…,sn Sorten sind.
- Der Argumentbereich von P ist die Menge Ms1 x … x Msn
- Das Tupel (s1,…,sn) ist der Argumenttyp von P.
- Die Zahl n > 0 ist die Stelligkeit von P.
- Ist P(a1,…,an) = W, so sagen wir: P trifft auf a1,…,an zu

Struktur: Blockwelt-Beispiel

Bereiche

alle Blöcke: a,b,c,d,e

alle Pfeiler, die aus den Blöcken gebaut werden können

alle Bögen, die gebaut werden können

Prädikate

• Block(x)	x ist ein Block
• Pfeiler(x)	x ist ein Pfeiler
• Bogen(u)	u ist ein Bogen
• auf-Boden(x)	x steht auf dem Boden
• liegt-auf(x,y)	x liegt auf y

Operationen

• aus-einem(x)	das Gebilde aus x allein
• aus-zwei(x,y)	das Gebilde aus x auf y
• aus-drei(x,u,v)	das Gebilde aus x auf u und v

Struktur: Logelei-Beispiel

Bereiche

alle Leute: Arno, Emil, Gustav, Fritz, Passant, Mann

Prädikate

• Wa(x)	x sagt die Wahrheit
• Tä(x)	x ist der Täter

Symbole & Interpretation

Logik:

Trennung von Namen und Bedeutung , d.h.

- führe Symbole , d.h. Zeichen und Wörter (ohne Bedeutung) ein
- ordne eine Bedeutung zu, d.h. definiere Wörter und interpretiere Zeichen

Bsp.:

- Datensymbole(Konstanten): Arno, Fritz, Gustav, a, b, …
- Sortensymbole: Pfeiler, Leute, …
- Operationssymbole: aus-zwei, aus-einem, …
- Aussagensymbole: Wa(Gustav), Pfeiler(a), …
- Prädikatsymbole: auf-Boden, Wa, …

Formalisieren:

übertrage möglichst viel, von dem, was man weiß, von der natürlichen Sprache in die Regeln und Definitionen des Kalküls

Bsp: Logelei

Symbole, Signatur, Interpretation

Definition 1.5.

- Eine Signatur ist eine Menge von Sortensymbolen sowie von Daten-,Operations-, Aussagen- und Prädikatsymbolen mit Argument- und Werttyp aus diesen Sorten.
- Datensymbole sind nullstellige Operationssymbole.
- Aussagensymbole sind nullstllige Prädikatsymbole.
- Eine Signatur, die nur Aussagensymbole enthält, heißt aussagenlogisch.
- Ist *M* eine Struktur, so heißt die Menge der Sorten-, Operations- und Prädikat-symbole, die zu ihrer Beschreibung benutzt werden, Signatur der Struktur *M*
- Ist Σ- eine Signatur, so heißt eine Struktur, die zu jeden Sorten-, Operations- und Prädikatsymbol von Σ eine Sorte, Operation oder Prädikat von gleichem Argument- und Werttyp enthält, eine Struktur der Signatur Σ.
- Die Abbildung, die jedem Symbol eine passende Bedeutung zuordnet, heißt Interpretation I. Man schreibt I(Σ) = *M*.

Signatur: Blockwelt

- Sortensymbole: Block, Pfeiler, Bogen
- Konstanten:

  a -> Block, b -> Block, … ,e -> Block
- Operationssymbole:

  aus-einem: Block -> Pfeiler

  aus-zwei: Block x Pfeiler -> Pfeiler

  aus-drei: Block x Pfeiler x Pfeiler -> Bogen
- Prädikatsymbole:

Block: Block -> {W,F}

Pfeiler: Pfeiler -> {W,F}

Bogen: Bogen -> {W,F}

auf-Boden: Block -> {W,F}

liegt-auf: Block x Pfeiler -> {W,F}

Signatur: Natürliche Zahlen

- Sortensymbole: N
- Konstanten:

  0 -> N, 1 -> N

- Operationssymbole:

  ´: N -> N   Nachfolger, einstellige Operation

  +: N x N -> N   Addition, zweistellige Operation

  *: N x N -> N   Multiplikation, zweistellige Operation

- Prädikatsymbole:

  =: N x N -> {W,F}

  <: N x N -> {W,F}

  Š: N x N -> {W,F}

Terme

Definition 1.6.

Σ-Terme der Sorte s werden definiert durch:

(1) Die Variablen und Konstanten der Sorte s sind Σ-Terme der Sorte s.

(2) Sind t1,…,tn Σ-Terme der Sorten s1,…,sn und ist f: s1,…,sn ->s ein Operationssymbol in Σ, so ist f(t1,…,tn) ein Σ-Term der Sorte s.

(3) Das sind alle Σ-Terme der Sorte s.

- Statt Σ-Terme sagt man auch Term zur Signatur Σ.
- Sind Signaturen und/oder Sorten unwichtig oder aus dem Kontext ersichtlich, dann können sie weggelassen werden, und man sagt einfach Term.
- Ein Term ist variablenfrei oder Grundterm, wenn er keine Variablen enthält.

Terme: Beispiele

Terme (mit Sorten in Klammern) sind für das Blockwelt-Beispiel

x,y	(Block)
u,v	(Pfeiler)
a(Block)	
aus-einem(x)	(Pfeiler)
aus-zwei(x,u)	(Pfeiler)
aus-drei(d,aus-zwei(b,u),aus-einem(c))	(Bogen)
aus-zwei(a,aus-zwei(b,aus-einem(c)))	(Pfeiler)

keine Terme sind

aus-drei(d,aus-einem(b))

aus-zwei(y,a)

aus-drei(aus-drei(e,aus-einem(c),aus-einem(d)),
aus-einem(a),aus-einem(b))

Formeln

Definition 1.7.

Σ-Formeln (für die offene Prädikatenlogik) werden definiert durch:

(1) Die Wahrheitswerte und die Aussagensymbole in Σ sind Σ-Formeln.

Sind t1,…,tn Σ-Terme der Sorten s1,…,sn und ist P: s1,…,sn ein Prädikatensymbol in Σ, so ist P(t1,…,tn) eine Σ-Formel.

(2) Sind A und B Formeln, so auch $\neg$ A, (A $\wedge$ B), (A $\vee$ B), (A $\rightarrow$ B), (A $\leftrightarrow$ B)

(3) Das sind alle Σ-Formeln.

- Ist die Signatur klar, dann sagt man einfach Formel.
- Eine Formel ist variablenfrei oder Grundformel, wenn sie keine Variablen enthält.
- Formeln, die nur Aussagensymbole enthalten, heißen aussagenlogisch.

Formeln: Beispiele

Formeln oder keine Formeln???

Betrachten Sie die Signatur Σ mit dem Sortensymbol '*sorte*',
der Konstanten *e->sorte*,
den einstelligen Operatoren *f(sorte) -> sorte* und *g(sorte) -> sorte*,
dem zweistelligen Operationssymbol *h(sorte,sorte) -> sorte,*
dem einstelligen Prädikatensymbol *P(sorte)*,
dem zweistelligen Prädikatensymbol *R(sorte,sorte)*.

x1, x2, y1, y2, z seien Variablen zur Sorte *sorte*.

- f(f(f(e)))
- (P(z) ∨ R(x1,x2))
- P(f(z))
- f(x)
- f(h(e,P(x1)))
- (P ∨ ¬ P)
- R(g(e),e)
- P(f(g(e)))
- R(f(g(x1,e),x2))
- ((P(e) ∧ R(e,e)) ∨ ¬ P(f(e)))
- (R(g(f(e)),f(f(e)))
- (P(z) → R(e,f(e,e))
- h(f(e),g(e))
- (P(e) ↔ P(f(g(e))))

Auswertung von variablenfreien Termen und Formeln

Definition 1.8.

Es sei *M* eine Struktur mit der Signatur Σ. Der Wert von variablenfreien Σ-Termen und Σ-Formeln wird definiert durch folgende Auswertungsregeln:

(1) Für jedes Operationssymbol f: s1,…,sn->s und Daten a1.…,an und b der Sorten s1,…,sn und s ersetzen wir f(a1,…,an) durch seinen Wert:

f(a1,…,an) ~> b, falls f(a1,…,an) = b in *M*.

Insbesondere für jede Konstante f ~> b, falls f = b in *M*.

(2) Für jedes Prädikatensymbol P: s1,…,sn->{W,F} und Daten a1.…,an der Sorten s1,…,sn entsprechend:

P(a1,…,an) ~> W, falls P auf a1,…,an in *M* zutrifft, P(a1,…,an) ~> F sonst.

Insbes. für jedes Aussagensymbol P ~> W, falls P in *M* zutrifft, P~> F sonst.

Ein Datum a, das sich für einen variablenfreien Term t bei der Auswertung in *M* ergibt, heißt (Daten-)Wert von t in*M* ; wir schreiben $wert_M$ (t) = a und sagen: t hat den Wert a in *M* oder t=a in *M*.

Ein Wahrheitswert, der sich für eine variablenfreie Formel A bei der Auswertung in *M* ergibt, heißt (Wahrheits-)Wert von A in*M* ; wir schreiben $wert_M$ (A) und sagen: A ist wahr (bzw. falsch) in *M* wenn $wert_M$ (A) = W (bzw. F) ist.

Auswertung: Beispiele

- **natürliche Zahlen**

  (3 + 4) * (7 + 2) ~>

  (3 * 4) < (3 + 2) ~>

- **Blockwelt**

  Bogen(aus-drei(d,aus-zwei(b,aus-einem(a)),aus-einem(c))) ~>

  aus-zwei(d,aus-zwei(a,aus-zwei(b,aus-einem(c)))) ~>

Substitution

Definition 1.9.

Es seien T ein Term und A eine Formel, es seien x1,…,xn Variablen der Sorten s1,…,sn und t1,…,tn Terme der Sorten s1,…,sn. Dann entsteht

- T(t1,…,tn) aus T(x1,…,xn), indem gleichzeitig die Terme t1,…,tn für die Variablen x1,…,xn überall in T eingesetzt (substituiert) werden.
- A(t1,…,tn) aus A(x1,…,xn), indem gleichzeitig die Terme t1,…,tn für die Variablen x1,…,xn überall in A eingesetzt (substituiert) werden.

Der Vorgang und die zugehörige Abbildung σ

$\sigma =$ {x1/t1,…,xn/tn}

mit der Terme in Terme und Formeln in Formeln überführt werden, heißt Einsetzung oder Substitution.Tσ bzw. Aσ heißt Instanz von T bzw. A

Achtung: es muß überall der gleiche Term eingesetzt werden.

Beispiele: Sei σ = {x/2, y/7}. Dann ist ((x+y)*(y+x))σ =

und (x+x)σ =

Gültige Formeln und Modelle

Definition 1.10.

Es seien A eine Formel mit den Variablen x1,…,xn und *M* eine Struktur passender Signatur. Die Formel A gilt in *M* (oder ist gültig in *M)*, wenn für alle Daten a1,…,an in *M* passender Sorten die Formel

$A\sigma$ mit $\sigma=\{x1/a1,\ldots,xn/an\}$

in *M* wahr ist. Eine Formelmenge X gilt in *M* (oder ist gültig in *M)*, wenn alle Formeln aus X in *M* gelten.

Gelten A bzw. X in *M* , so heißt *M* ein Modell von A bzw. X.

Gültige Formeln & Modelle: Beispiele

Affe & Banane: A1-A13

A1	$Arme(x) \land Nah(x,y) \rightarrow Reichen(x,y)$				
A2	$Auf(x,y) \land Unter(y,Bananen) \land Hoch(y) \rightarrow Nah(x,Bananen)$				
A3	$In(x) \land In(y) \land In(z) \land Schieben(x,y,z) \rightarrow Nah(z, Boden) \lor Unter(y,z)$				
A4	$Steigen(x,y) \rightarrow Auf(x,y)$				
A5	Arme(Affe)	A6	Hoch(Kiste)		
A7	In(Affe)	A8	In(Bananen)	A9	In(Kiste)
A10	Schieben(Affe,Kiste,Bananen)			A11	$\neg$Nah(Bananen,Boden)
A12	Steigen(Affe,Kiste)			A13	Reichen(Affe,Bananen)

gelten in der Struktur

- Bereich: Affe, Kiste, Bananen
- Prädikate: Arme(x) Nah(x,y) Reichen(x,y)
  Auf(x,y) Unter(x,y) Hoch(x)
  In(x) Steigen(x,y) Schieben(x,y,z)

d.h. diese Struktur ist ein Modell der Formeln.

Quantorenlogik: Beispiel(1)

Euklidische Ebene (Geometrie)

E1 Auf jeder Geraden liegt mindestens ein Punkt.

E2 Jeder Punkt liegt auf einer Geraden.

E3 Zu jeder Geraden und jedem Punkt darauf gibt es einen weiteren Punkt darauf.

E4 Durch zwei verschiedene Punkte ist die Gerade, die sie enthält, eindeutig bestimmt.

E5 Zwei verschiedene Geraden haben höchstens einen Schnittpunkt.

E6 Liegt ein Punkt zwischen zwei anderen, so ist er verschieden von ihnen.

E7 Zwischen zwei verschiedenen Punkten liegt ein dritter.

E8 Liegen zwei Punkte auf einer Geraden, und ein dritter dazwischen, so liegt er auch auf der Geraden.

E9 Zwei Geraden sind genau dann parallel, wenn sie sich nicht schneiden.

E10 Enthält eine Gerade einen Punkt nicht, so gibt es eine Parallele dazu durch den Punkt.

E11 Die Parallele zu einer Geraden durch einen nicht auf ihr liegenden Punkt ist eindeutig bestimmt.

Quantorenlogik: Beispiel(2)

Einführung von Quantoren

- Allquantor: $\forall$ x A für: auf alle x trifft A zu
- Existenzquantor: $\exists$ x A für: es gibt ein x, auf das A zutrifft

Formalisierung des Beispiels:

- Sorten: 'Gerade', 'Punkt'
- Variablen: p,q,r (für Punkt), G,H,I (für Gerade)
- Prädikatsymbole

Punkt Gerade — p liegt auf G: p G

Gerade $\parallel$ Gerade — G ist parallel zu H: G $\parallel$ H

Punkt Punkt Punkt — r zwischen p und q: p r q

Punkt = Punkt; Gerade=Gerade — Gleichheit

Quantorenlogik: Beispiel(3)

Euklidische Ebene (Geometrie)

E1 $\forall G \, \exists p \, (-p- G)$ Auf jeder Geraden liegt mindestens ein Punkt.

E2 $\forall p \, \exists G \, (-p- G)$ Jeder Punkt liegt auf einer Geraden.

E3 $\forall G \, \forall p \, (-p- G \rightarrow \exists q \, (q \circ p \wedge -q- G))$

Zu jeder Geraden und jedem Punkt darauf gibt es einen weiteren Punkt darauf.

E4 $\forall p \, \forall q \, \forall G \, \forall H \, (p \circ q \wedge -p- G \wedge -q- G \wedge -p- H \wedge -q- H \rightarrow G = H)$

Durch zwei verschiedene Punkte ist die Gerade, die sie enthält, eindeutig bestimmt.

…

E7 $\forall p \, \forall q \, (p \circ q \rightarrow \exists r \; -p-r-q-)$

Zwischen zwei verschiedenen Punkten liegt ein dritter.

…

E10 $\forall p \, \forall G \, (\neg \; -p- G \rightarrow \exists H \, (H \parallel G \wedge -p- H))$

Enthält eine Gerade einen Punkt nicht, so gibt es eine Parallele dazu durch den Punkt.

…

Quantorenlogik

Definition 1.11.

Es sei Σ eine Signatur und die Σ-Terme wie in Definition 1.6. Die Formeln zur Signatur Σ oder Σ-Formeln werden definiert durch:

(1) Die Wahrheitswerte W, F und die Aussagensymbole in Σ sind Σ-Formeln.

Sind t1,…,tn Σ-Terme der Sorten s1,…,sn und ist P: s1,…,sn ein Prädikatensymbol in Σ, so ist P(t1,…,tn) eine Σ-Formel.

(2) Sind A und B Formeln und x eine Variable, so sind

$\neg A, (A \wedge B), (A \vee B), (A \rightarrow B), (A \leftrightarrow B), \forall x A, \exists x A$

Σ-Formeln.

- $\forall x$, $\exists x$ werden als Quantoren bezeichnet.
- in $\forall x A$, $\exists x A$ ist A der Bereich des Quantors
- Klammerkonvention: $\forall x A$, $\exists x A$ binden ebenso stark wie $\neg$

Freie und gebundene Variablen

Definition 1.12.

- Wenn eine Variable x an einer bestimmten Stelle in einer Formel A im Bereich eines Quantors $\forall$ x oder $\exists$ x steht, ist x in A an dieser Stelle gebunden; sonst ist sie in A an dieser Stelle frei.
- Die Variable x kommt in A frei (gebunden) vor, wenn sie in A an einer Stelle frei (gebunden) ist. Die freien Variablen von A sind die Variablen, die in A frei vorkommen.
- Eine Formel ohne freie Variablen heißt geschlossen.

Beispiele:

- in — p — G $\to \exists$ p (— p — G)

  kommt p links von $\to$ frei, rechts davon gebunden vor; G kommt frei vor
- in $\neg$ — p — G $\to \exists$ H (H $\parallel$ G $\wedge$ — p — H)

  kommen G und p frei vor; H kommt gebunden vor

Substitution

Definition 1.13.

Wie Definition 1.9. Zusätzlich gilt folgende Rekursion:

- (Qx B){x/t} = QxB , (Qy B){x/t} = Qy (B{x/t}) für x°y, wobei Q ein Quantor ist.

Für freie Variablen hat die Substitution die gleichen Eigenschaften wie bisher.

Beispiel

- (—p— G → ∃ p (—p— G)){p/p_3, G/G_126} =

  (—p_3— G_126 → ∃ p (—p— G_126))

Auswertung (Quantorenlogik)

Definition 1.14.

Es sei *M* eine Struktur mit der Signatur Σ.

- Für Terme, atomare Formeln und aussagenlogische Verknüpfungen gelten die Auswertungsregeln wie in Definition 1.8.
- Zur Auswertung von geschlossenen Formeln Qx A gelten folgende Regeln:

  $\forall$ x A ~> W, falls sich A{x/a} für alle Daten a zu W auswerten läßt; $\forall$ x A ~> F sonst

  $\exists$ x A ~> W, falls sich A{x/a} für irgendein Datum a zu W auswerten läßt; $\exists$ x A ~> F sonst

- Ein Wahrheitswert, der sich aus einer geschlossenen Formel A bei der Auswertung in *M* ergibt, heißt (Wahrheits-)Wert von A in*M* ; wir schreiben wert_M (A) und sagen: A ist wahr (bzw. falsch) in *M* wenn wert_M (A) = W (bzw. F) ist.
- Eine Formel A gilt in*M* (oder ist gültig in *M)*, wenn sie für alle Daten, die für ihre freien Variablen eingesetzt werden, in *M* wahr ist. Eine Formelmenge X gilt in*M* (oder ist gültig in*M)*, wenn alle Formeln aus X in *M* gelten. *M* ist dann ein Modell von A bzw. X.