Mathematik und Logik

2007W

Institut für Algebra
Johannes Kepler Universität Linz

Vorlesung im 2007W

http://www.algebra.uni-linz.ac.at/Students/Win/ml

Inhalt

Aussagen

Die mathematische Logik verwendet mathematische Methoden, um das logische Denken formal zu beschreiben.

- Populäre Definition: Eine Aussage ist ein Satz, der entweder falsch oder wahr ist.
- Problem: Wie definiert man *wahr* und *falsch*?
- Ein Beweis stellt sicher, daß eine Aussage wahr ist.
- Definition: Eine Aussage ist eine Konstruktion, durch welche festgelegt wird, wie ihre Beweise zu konstruieren sind.
- Eine Aussage, die wir nicht beweisen können, muß nicht unbedingt falsch sein.

Definition der Implikation, $\Rightarrow$

Formation

Sind P und Q Aussagen, dann bezeichnet $P \Rightarrow Q$ ebenfalls eine Aussage, die Implikation von P und Q.

Introduktion

Um $P \Rightarrow Q$ zu beweisen, muß man Q beweisen, wobei man einen Beweis von P voraussetzen darf.

Elimination

Hat man einen Beweis von $P \Rightarrow Q$, so reicht ein Beweis von P, um auch Q zu beweisen.

Schlussregeln

$$\frac{\begin{array}{c}[P]\\ \vdots\\ Q\end{array}}{P \Rightarrow Q}\Rightarrow\mathcal{I} \qquad \frac{P \Rightarrow Q \quad P}{Q}\Rightarrow\mathcal{E}$$

Definition der Konjuktion, $\wedge$

Formation

Sind P und Q Aussagen, dann bezeichnet $P \wedge Q$ ebenfalls eine Aussage, die Konjunktion von P und Q.

Introduktion

Um $P \wedge Q$ zu beweisen, muß man sowohl P als auch Q beweisen.

Elimination

Hat man einen Beweis von $P \wedge Q$ so auch einen Beweis von P, und auch einen Beweis von Q.

Schlussregeln

$$\frac{P \quad Q}{P \wedge Q}\wedge\mathcal{I} \qquad \frac{P \wedge Q}{P}\wedge\mathcal{E}_0 \qquad \frac{P \wedge Q}{Q}\wedge\mathcal{E}_1$$

Kommutativität der Konjunktion

Satz

Die logische Konjunktion ist kommutativ, d.h. die Aussage

$$A \wedge B \Rightarrow B \wedge A$$

ist allgemeingültig.

Beweis.

$$\dfrac{\dfrac{\boxed{\dfrac{\dfrac{A \wedge B}{B}\wedge\mathcal{E}_1 \quad \dfrac{A \wedge B}{A}\wedge\mathcal{E}_0}{B \wedge A}\wedge\mathcal{I}}}{A \wedge B}}{A \wedge B \Rightarrow B \wedge A}\Rightarrow\mathcal{I}$$

□

Definition der Äquivalenz, $\Longleftrightarrow$

Notation

Die logische Äquivalenz wird mit $P \iff Q$ bezeichnet und ist lediglich eine Abkürzung für $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$.

Bemerkung

Zwei Aussagen sind äquivalent wenn sie vom logischen Standpunkt aus betrachtet gleichwertig sind.

Definition der Disjunktion, $\vee$

Formation

Sind P und Q Aussagen, dann bezeichnet $P \vee Q$ ebenfalls eine Aussage, die Disjunktion von P und Q.

Introduktion

Um $P \vee Q$ zu beweisen, genügt es, P zu beweisen, oder Q zu beweisen.

Elimination

Folgt irgendeine Aussage R sowohl aus P als auch aus Q, dann folgt sie auch aus $P \vee Q$ (Beweis durch Fallunterscheidung).

Schlussregeln

$$\frac{P}{P \vee Q}\vee\mathcal{I}_0 \qquad \frac{Q}{P \vee Q}\vee\mathcal{I}_1 \qquad \frac{P \Rightarrow R \quad Q \Rightarrow R}{P \vee Q}\vee\mathcal{E} \; R$$

Allquantor, $\forall$

- Sei X ein Datentyp, und $P[x]$ für jedes $x \in X$ eine Aussage. Dann bezeichnet $\forall_{x \in X} P[x]$ eine All-Aussage.
- Die All-Aussage drückt eine universelle Quantifizierung aus.
- Ein Beweis von $\forall_{x \in X} P[x]$ konstruiert für jedes beliebige $x \in X$ einen Beweis von $P[x]$.
- Praktisch: Es sei $\forall_{x \in X} P[x]$ zu beweisen.
  Vorgangsweise: Annahme $x \in X$; beweise $P[x]$.
- Hängt $P[x]$ nicht von x ab, dann liegt eine normale Implikation vor: $\forall_{x \in X} P$ ist dasselbe wie $X \to P$.

Allquantor, Introduktion und Elimination

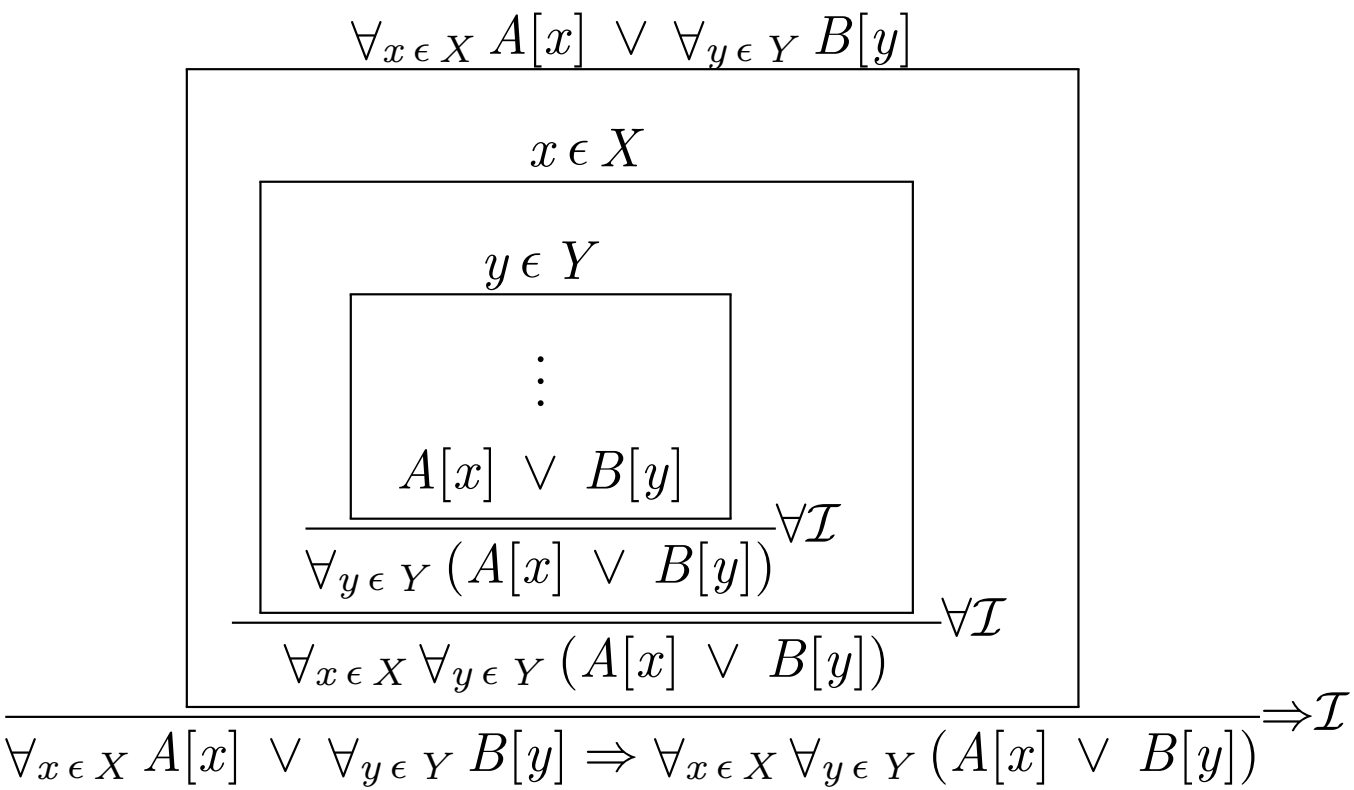
- Um $\forall_{x \in X} P[x]$ zu beweisen, muß man $P[x]$ für ein beliebiges $x \in X$ beweisen.
- $\forall$-Introduktion:

$$\dfrac{\begin{array}{c}[x \in X] \\ \vdots \\ P[x]\end{array}}{\forall_{x \in X} P[x]} \forall\mathcal{I}$$

- Wurde $\forall_{x \in X} P[x]$ bewiesen, und ist $a \in X$, dann hat man einen Beweis von $P[a]$.
- $\forall$-Elimination:

$$\dfrac{\forall_{x \in X} P[x] \quad a \in X}{P[a]} \forall\mathcal{E}$$

Allquantor, Beispiel



Allquantor: Beispiel (Fortsetzung)

Annahmen: $\forall_{x \,\epsilon\, X}\, A[x] \;\vee\; \forall_{y \,\epsilon\, Y}\, B[y], \qquad x \,\epsilon\, X, \quad y \,\epsilon\, Y$

Zu beweisen:

$$\dfrac{\begin{array}{cc}\overset{\forall_{x \,\epsilon\, X}\, A[x]}{\boxed{\begin{array}{c}\vdots\\ A[x] \;\vee\; B[y]\end{array}}} & \overset{\forall_{y \,\epsilon\, Y}\, B[y]}{\boxed{\begin{array}{c}\vdots\\ A[x] \;\vee\; B[y]\end{array}}}\end{array}}{\forall_{x \,\epsilon\, X}\, A[x] \;\vee\; \forall_{y \,\epsilon\, Y}\, B[y] \Rightarrow A[x] \;\vee\; B[y]}\vee\mathcal{E}$$

Die beiden Fälle:

$$\dfrac{\dfrac{\forall_{x \,\epsilon\, X}\, A[x] \quad x \,\epsilon\, X}{A[x]}\forall\mathcal{E}}{A[x] \;\vee\; B[y]}\vee\mathcal{I}_0 \qquad \dfrac{\dfrac{\forall_{y \,\epsilon\, Y}\, B[y] \quad y \,\epsilon\, Y}{B[y]}\forall\mathcal{E}}{A[x] \;\vee\; B[y]}\vee\mathcal{I}_1$$

Existenzquantor, $\exists$

- Sei X ein Datentyp, und $P[x]$ für jedes $x \in X$ eine Aussage. Dann bezeichnet $\exists_{x \in X} P[x]$ eine Existenz-Aussage.
- Die Existenzaussage drückt eine existenzielle Quantifizierung aus.
- Ein Beweis von $\exists_{x \in X} P[x]$ konstruiert ein $a \in X$ und einen Beweis von $P[a]$.
- Praktisch: Es sei $\exists_{x \in X} P[x]$ zu beweisen.
  Vorgangsweise: Man wählt ein passendes $a \in X$, und versucht damit $P[a]$ zu beweisen.
- Hängt $P[x]$ nicht von x ab, dann liegt eine normale Konjunktion vor: $\exists_{x \in X} P$ ist dasselbe wie $X \wedge P$.

Existenzquantor: Introduktion und Elimination

- Um $\exists_{x \,\epsilon\, X}\, P[x]$ zu beweisen, muß man ein $a \,\epsilon\, X$ finden und damit $P[a]$ beweisen.
- $\exists$-Introduktion:
$$\frac{a \,\epsilon\, X \quad P[a]}{\exists_{x \,\epsilon\, X}\, P[x]}\exists\mathcal{I}$$
- Wurde $\exists_{x \,\epsilon\, X}\, P[x]$ bewiesen, so kann man für's weitere annehmen, daß es ein soches Objekt gibt.
- $\exists$-Elimination:

$$\frac{\exists_{x \,\epsilon\, X}\, P[x] \qquad \begin{array}{c} y \,\epsilon\, X \quad P[y] \\ \vdots \\ Q \end{array}}{Q}\exists\mathcal{E}$$

Implikation, $\Rightarrow$

- Schlußregeln:

$$\frac{\begin{array}{c}x \in P\\ \boxed{\begin{array}{c}\vdots\\ t[x] \in Q\end{array}}\end{array}}{x \mapsto t[x] \in P \Rightarrow Q}\Rightarrow\mathcal{I} \qquad \frac{f \in P \Rightarrow Q \quad x \in P}{fx \in Q}\Rightarrow\mathcal{E}$$

- Der Beweis einer Implikation ist ein Algorithmus,
- der für jeden Input vom Typ P einen Output vom Typ Q liefert.
- Funktionsdatentyp: Schreibweise: $P \to Q$ oder Q^P.
- Konstruktor: Abstraktion: $(\mapsto)$;
- Selektor: Funktionsanwendung: apply.

Logische Konjunktion, $\wedge$

- Introduktion und Elimination

$$\frac{x \,\epsilon\, P \quad y \,\epsilon\, Q}{(x, y) \,\epsilon\, P \wedge Q} \wedge\mathcal{I} \qquad \frac{z \,\epsilon\, P \wedge Q}{\mathrm{fst}\, z \,\epsilon\, P} \wedge\mathcal{E}_0 \qquad \frac{z \,\epsilon\, P \wedge Q}{\mathrm{snd}\, z \,\epsilon\, Q} \wedge\mathcal{E}_1$$

- Ein Beweis der Konjunktion $P \wedge Q$ ist ein Paar,
- dessen Komponenten die Typen P bzw. Q haben.
- Verbunddatentyp (Direktes Produkt): $P \times Q$.
- Konstruktor: $(,) \,\epsilon\, P \to Q \to P \times Q$;
- Selektoren: $\mathrm{fst} \,\epsilon\, P \times Q \to P$, $\mathrm{snd} \,\epsilon\, P \times Q \to Q$.

Kommutativität der Konjunktion

Satz

$A \wedge B \Rightarrow B \wedge A.$

- Beweis:

$$\dfrac{\dfrac{\dfrac{c \in A \wedge B}{\mathrm{snd}\, c \in B}\wedge\mathcal{E}_1 \qquad \dfrac{c \in A \wedge B}{\mathrm{fst}\, c \in A}\wedge\mathcal{E}_0}{(\mathrm{snd}\, c, \mathrm{fst}\, c) \in B \wedge A}\wedge\mathcal{I}}{c \mapsto (\mathrm{snd}\, c, \mathrm{fst}\, c) \in A \wedge B \Rightarrow B \wedge A}\Rightarrow\mathcal{I}$$

- $\mathrm{commute} \in A \times B \to B \times A,\ c \mapsto (\mathrm{snd}\, c, \mathrm{fst}\, c),$
- Intuitiver: $(a, b) \mapsto (b, a).$
- Äquivalenz:
  $\big(c \mapsto (\mathrm{snd}\, c, \mathrm{fst}\, c), c \mapsto (\mathrm{snd}\, c, \mathrm{fst}\, c)\big) \in A \wedge B \Leftrightarrow B \wedge A.$

Logische Disjunktion, $\vee$

- Introduktion und Elimination

$$\frac{x \,\epsilon\, P}{\text{Left}\; x \,\epsilon\, P \;\vee\; Q}\vee\mathcal{I}_0 \qquad \frac{y \,\epsilon\, Q}{\text{Right}\; y \,\epsilon\, P \;\vee\; Q}\vee\mathcal{I}_1$$

$$\frac{f \,\epsilon\, P \Rightarrow R \quad g \,\epsilon\, Q \Rightarrow R}{\text{either}\; f\; g \,\epsilon\, P \;\vee\; Q \Rightarrow R}\vee\mathcal{E}$$

- Ein Beweis der Disjunktion $P \vee Q$ ist einer von P oder von Q, *und als solcher gekennzeichnet.*
- Disjunkte Vereinigung (Direkte Summe): $P + Q$.
- Konstruktoren: $\text{Left} \,\epsilon\, P \to P + Q$, $\text{Right} \,\epsilon\, Q \to P + Q$;
- Selektor: $\text{either} \,\epsilon (P \to R) \to (Q \to R) \to (P + Q \to R)$.

Beispiel: $A \vee (B \wedge C) \Rightarrow A \vee B$

$$\dfrac{\dfrac{\boxed{\dfrac{\overset{a \in A}{\vdots}}{\text{Left } a \in A \vee B}\vee\mathcal{I}_0}}{a \mapsto \text{Left } a \in A \Rightarrow A \vee B}\Rightarrow\mathcal{I} \qquad \dfrac{\boxed{\dfrac{\dfrac{\overset{y \in B \wedge C}{\vdots}}{y \in B \wedge C}\wedge\mathcal{E}_1}{\text{fst } y \in B}\vee\mathcal{I}_1}}{y \mapsto \text{Right}(\text{fst } y) \in B \wedge C \Rightarrow A \vee B}\Rightarrow\mathcal{I}}{\text{either}(a \mapsto \text{Left } a)\,(y \mapsto \text{Right}(\text{fst } y)) \in A \vee (B \wedge C) \Rightarrow A \vee B}=$$

Mit

$$f \in A \vee (B \wedge C) \Rightarrow A \vee B$$
$$g \in A \vee (B \wedge C) \Rightarrow A \vee C$$

erhalten wir auch:

$$(f, g) \in A \vee (B \wedge C) \Rightarrow (A \vee B) \wedge (A \vee C)$$

Es gibt auch: $h \in (A \vee B) \wedge (A \vee C) \Rightarrow A \vee (B \wedge C)$

Curry-Howard-Isomorphismus

- Eine Aussage legt den Datentyp ihrer Beweise fest.
- Ein Datentyp entspricht der Aussage, daß es ein Objekt dieses Typs gibt.
- Jeder Algorithmus, der ein Objekt eines bestimmten Datentyps konstruiert, ist ein Beweis, daß es ein solches gibt.
- Aussagen entsprechen Programmspezifikationen.
- Beweise entsprechen Programmen.
- Man kann Aussagen beweisen, indem man ein Objekt vom passenden Typ konstruiert.
- Aus mathematischen Beweisen lassen sich verifizierte Programme extrahieren.
- *Fehlerfreie Software beliebiger Komplexität ist möglich.*