Mathematik und Logik

Franz Binder

Institut für Algebra
Johannes Kepler Universität Linz

Vorlesung im 2006W
`http://www.algebra.uni-linz.ac.at/Students/Win/ml`

Inhalt

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

$$\dfrac{\begin{array}{c}\boxed{\begin{array}{c}P\\ \vdots\\ Q\end{array}}\end{array}}{P \Rightarrow Q}\Rightarrow\mathcal{I} \qquad\qquad \dfrac{P \Rightarrow Q \quad P}{Q}\Rightarrow\mathcal{E}$$

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

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

Kommutativität der Konjunktion

Satz

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

$$A \wedge B \implies B \wedge A$$

ist allgemeingültig.

Beweis.

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

□

Definition der Äquivalenz, $\iff$

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

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

Implikation, $\Rightarrow$

- Schlußregeln:

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

- 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

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

- 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: $(,)\colon P \to Q \to P \times Q$;
- Selektoren: $\mathsf{fst}\colon P \times Q \to P$, $\mathsf{snd}\colon P \times Q \to Q$.

Kommutativität der Konjunktion

Satz
$A \wedge B \implies B \wedge A$.

- Beweis:

$$\cfrac{\cfrac{\boxed{\cfrac{\cfrac{c\colon A \wedge B}{\mathsf{snd}\, c\colon B}\wedge\mathcal{E}_1 \quad \cfrac{c\colon A \wedge B}{\mathsf{fst}\, c\colon A}\wedge\mathcal{E}_0}{(\mathsf{snd}\, c, \mathsf{fst}\, c)\colon B \wedge A}\wedge\mathcal{I}}}{c \mapsto (\mathsf{snd}\, c, \mathsf{fst}\, c)\colon A \wedge B \implies B \wedge A}\Rightarrow\mathcal{I}}{c\colon A \wedge B}$$

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

Logische Disjunktion, $\vee$

- Introduktion und Elimination

$$\frac{x\colon P}{\mathsf{Left}\,x\colon P\vee Q}\vee\mathcal{I}_0 \qquad \frac{y\colon Q}{\mathsf{Right}\,y\colon P\vee Q}\vee\mathcal{I}_1$$

$$\frac{f\colon P\Rightarrow R \quad g\colon Q\Rightarrow R}{\mathsf{either}\,f\,g\colon P\vee Q\Rightarrow R}\vee\mathcal{E}$$

- 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: $\mathsf{Left}\colon P\to P+Q$, $\mathsf{Right}\colon Q\to P+Q$;
- Selektor: $\mathsf{either}\colon (P\to R)\to(Q\to R)\to(P+Q\to R)$.

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

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

Mit

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

erhalten wir auch:

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

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

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.*