Script generated by TTT

Title: Esparza: Diskrete Strukturen (29.10.2013)

Date: Tue Oct 29 13:48:02 CET 2013

Duration: 83:23 min

Pages: 49

Aufgabe 2.4

- Zu zeigen: $\mathbb{N}^*$ ist abzählbar
- Korrekte Lösungen von:
  - Dario Be
  - Rudolf Hattenkofer
  - Jana Kuzmanova
  - Moritz Pfeiffer
  - Lilian Qian
  - Maximilian Schmidt
  - Martin Wauligmann
  - Sebastian Weiß

- Lösung 1:
  - Liste für $n = 0,1,2,\dots$ alle Tupeln, deren Summe n ist.
- Lösung 2:
  - Sei $X_{i,j} = \{1,2,\dots,i\}^j$
  - Liste $X_{1,1}, X_{1,2}, X_{2,1}, X_{1,3}, \dots$

$X_{4,3}$

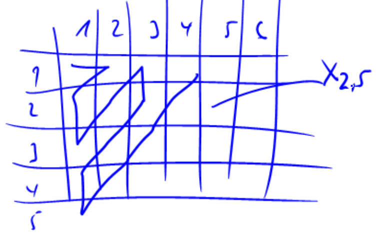
$(2, 3, 4)$

- Lösung 1:
  - Liste für $n = 0,1,2,\dots$ alle Tupeln, deren Summe n ist.
- Lösung 2:
  - Sei $X_{i,j} = \{1,2,\dots,i\}^j$
  - Liste $X_{1,1}, X_{1,2}, X_{2,1}, X_{1,3}, \dots$

$X_{4,3}$

$X_{5,3}$

$(2, 3, 4)$



3

Vorlesung Diskrete Strukturen WS 13/14
Prof. Dr. J. Esparza – Institut für Informatik, TU München

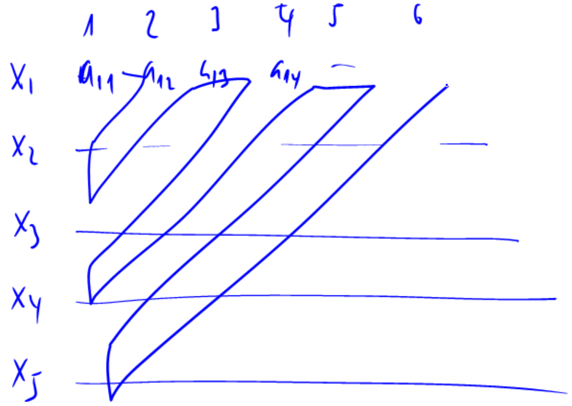
- Lösung 1:
  - Liste für $n = 0,1,2,\ldots$ alle Tupeln, deren Summe n ist.
- Lösung 2:
  - Sei $X_{i,j} = \{1, 2, \ldots, i\}^j$
  - Liste $X_{1,1}, X_{1,2}, X_{2,1}, X_{1,3}, \ldots$
- Lösung 3:
  - Wenn A abzählbar, dann A^k abzählbar für jedes $k \geq 1$.
  - Wenn $X_1, X_2, \ldots$ abzählbar, dann $\bigcup_{i=1}^{\infty} X_i$ abzählbar.

A

$A \times A$

A^3 $\underbrace{(A \times A)}\times A$

$A \times A \times \ldots \times A$



- Lösung 1:
  - Liste für $n = 0,1,2,\ldots$ alle Tupeln, deren Summe n ist.
- Lösung 2:
  - Sei $X_{i,j} = \{1, 2, \ldots, i\}^j$
  - Liste $X_{1,1}, X_{1,2}, X_{2,1}, X_{1,3}, \ldots$
- Lösung 3:
  - Wenn A abzählbar, dann A^k abzählbar für jedes $k \geq 1$.
  - Wenn $X_1, X_2, \ldots$ abzählbar, dann $\bigcup_{i=1}^{\infty} X_i$ abzählbar.

A

$A \times A$

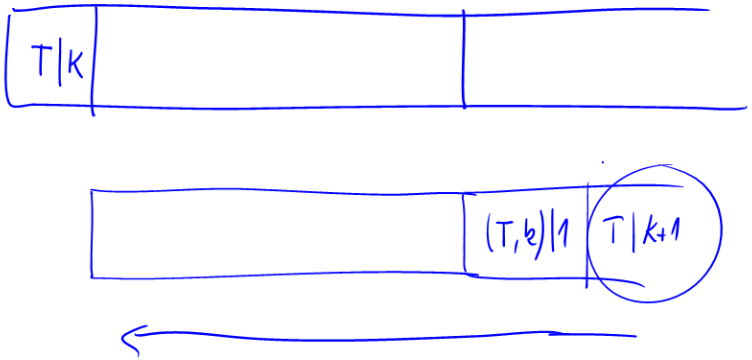
A^3 $\underbrace{(A \times A)}\times A$

$A \times A \times \ldots \times A$

4

Vorlesung Diskrete Strukturen WS 13/14
Prof. Dr. J. Esparza – Institut für Informatik, TU München

- Lösung 4:
  - Die Kinder eines Tupels $(n_1, \ldots, n_k)$ sind die Tupeln $(n_1, \ldots, n_k, 1), (n_1, \ldots, n_k, 2), \ldots$
  - Ein Tupel „will" seine Kinder der Reihe nach zur Welt bringen.
  - Die Tupeln bilden eine Schlange. Am Anfang steht nur das leere Tupel da.
  - In jedem Schritt gebärt das erste Tupel aus der Schlange sein nächstes Kind uns beide stellen sich ans Ende der Schlange (Kind vor Mutter).
  - Wenn ein Tupel geboren wird schreit es: „Hurrah, ich bin da"



Kapitel II – Grundlagen; Logik

- Noch zwei korrekte Inferenzen

  Wenn

  Anna ein Enkelkind hat

  dann

  ist Anna eine Mutter

Kapitel II – Grundlagen; Logik

- Noch zwei korrekte Inferenzen

  Wenn

  Anna ein Enkelkind hat

  dann

  ist Anna eine Mutter

  – Die Inferenz kann nur als korrekt erkannt werden, wenn man Wissen über die Bedeutung von „Mutter“ und „Enkelkind“ hat.

Kapitel II – Grundlagen; Logik

- Noch zwei korrekte Inferenzen

  Wenn

  alle Guldräber untreßig und alle Untreßigen filzig sind

  dann

  sind alle Guldräber filzig.

  – Diese Inferenz erkennen wir als korrekt, auch wenn wir nicht wissen, was „ Guldräber“, „untreßig“, oder „filzig“ bedeutet !!

  – Die Inferenz ist eine Instanz von:

  Wenn (alle X sind Y) und (alle Y sind Z) dann (alle X sind Z)

Kapitel II – Grundlagen; Logik

- Noch zwei korrekte Inferenzen

  Wenn

  Anna ein Enkelkind hat

  dann

  ist Anna eine Mutter

  – Die Inferenz kann nur als korrekt erkannt werden, wenn man Wissen über die Bedeutung von „Mutter“ und „Enkelkind“ hat.

Kapitel II – Grundlagen; Logik

- Logische Inferenzen (informell)
  – Eine logische Inferenz ist eine Inferenz mit „Variablen“ (Platzhalter für Aussagen oder Objekte).

  Achtung: diese Terminologie ist nicht Standard!

  – Eine logische Inferenz ist korrekt, wenn alle ihre Instanzen korrekt sind. Logiker sagen: die Inferenz ist (allgemein)gültig oder eine Tautologie.
    - „Wenn (alle X sind Y) und (alle Y sind Z) dann (alle X sind Z)“ ist gültig.
    - „Wenn (alle X sind Y) dann (alle Y sind X)“ ist nicht gültig.
    - „Wenn A und B und (wenn A dann C) dann B und C“ ist gültig.

Kapitel II – Grundlagen; Logik

- Die Logik untersucht die gültigen Inferenzen
  – Wie kann man sie (automatisch) erkennen?
  – Welche sind die nützlichsten?
- Die Logik stellt die Basis der mathematischen Beweise.
- Praktische Anwendungen der Logik finden sich in zahlreichen Gebieten der Informatik, z.B. Datenbanken, Programmverifikation, Korrektheitsbeweise, Systemsicherheit, künstliche Intelligenz und vielen mehr.

Kapitel II – Grundlagen; Logik

- Die Logik untersucht die gültigen Inferenzen
  – Wie kann man sie (automatisch) erkennen?
  – Welche sind die nützlichsten?
- Die Logik stellt die Basis der mathematischen Beweise.
- Praktische Anwendungen der Logik finden sich in zahlreichen Gebieten der Informatik, z.B. Datenbanken, Programmverifikation, Korrektheitsbeweise, Systemsicherheit, künstliche Intelligenz und vielen mehr.

14

Vorlesung Diskrete Strukturen WS 13/14
Prof. Dr. J. Esparza – Institut für Informatik, TU München

Kapitel II – Grundlagen; Logik

- Es gibt zahlreiche Logiken
  - Aussagenlogik, Prädikatenlogik, Temporalogik, deontische Logik …
- Verschiedene Logiken studieren die gültigen Inferenzen in verschiedenen Sprachfragmenten:
  - Aussagenlogik: „und“, „oder“, „nicht“, „wenn … dann„
    - „Wenn X und Y dann Y“ (Platzhalter für Aussagen)
  - Syllogistische Logik: „alle“, „einige“, „keine“
    - „Wenn alle A sind B und kein B ist ein C dann ist kein A ein C“ (Platzhalter für Objekte)
  - Temporallogik: Aussagenlogik + „heute“, „morgen“, „irgendwann“, „nie“
    - „Wenn heute A gilt, dann war gestern wahr, dass morgen A gelten wird“.

Kapitel II – Grundlagen; Logik

- Aussagenlogik (propositional logic)
  - Aussagen werden aus einer vorgegebene Menge von atomaren Aussagen oder Aussagenvariablen (Platzhalter für Aussagen) mit Hilfe der Operatoren (Konnektoren, Junktoren) „und“, „oder“, „nicht“ und „wenn … dann“ gebaut.
  - Atomare Aussagen sind wahr oder falsch, nie wahr und falsch, keines von beiden oder etwas “dazwischen”.
  - Die Grundlagen der Aussagenlogik wurden von George Boole („The Laws of Thought“ 1854) entwickelt. Man spricht deswegen auch von der Booleschen Logik.

Kapitel II – Grundlagen; Logik

- Aussagenlogik formal
  - Die Aussagenlogik (wie jede Logik) bildet eine formale Sprache.
  - Eine formale Sprache wird definiert durch ihre Syntax und ihre Semantik.
    - Die Syntax der Sprache legt durch Regeln fest, welche Zeichenketten wohlgeformte Ausdrücke sind.
      Die wohlgeformten Ausdrücke einer Logik heißen Formeln.
    - Die Semantik legt die Bedeutung der Formeln fest.
      Eine formale Semantik ordnet jeden Ausdruck einem mathematischen Objekt zu, welchem als Bedeutung des Ausdrucks interpretiert wird.

Kapitel II – Grundlagen; Logik

- Formale Syntax.
  - Eine formale Syntax besteht aus einem Vokabular und einer Menge von Formationsregeln.
  - Das Vokabular legt fest, welche Zeichen in Ausdrücken vorkommen dürfen
  - Die Formationsregeln legen fest, welche Zeichenketten über dem Vokabular zulässig oder ‘wohlgeformt’ sind und welche nicht.

Kapitel II – Grundlagen; Logik

- Syntax der Aussagenlogik: Formationsregeln

Regel 0: **true** und **false** sind Formeln.

Regel 1: eine Aussagenvariable ist eine Formel.

Regel 2: ist F eine Formel, dann ist auch $\neg F$ eine Formel.

Regel 3: sind F und G Formeln, dann sind

a. $(F \wedge G)$
b. $(F \vee G)$
c. $(F \rightarrow G)$

ebenfalls Formeln.

Regel 4: Ein Ausdruck ist nur dann eine Formel, wenn er durch Anwendung der obenstehenden Regeln konstruiert werden kann.

Kapitel II – Grundlagen; Logik

- Syntax der Aussagenlogik: Formationsregeln
  - Es lassen sich also durch Anwendung von logischen Operatoren (Konnektoren) auf Formeln neue Formeln bilden.
  - Ein Operator kombiniert einen oder mehrere Operanden zu einem komplexeren Ausdruck.
  - Monadische/unäre Operatoren haben ein Argument (z.B. $\neg F$), dyadische/binäre Operatoren haben zwei

Kapitel II – Grundlagen; Logik

- Syntax der Aussagenlogik: Formationsregeln

Regel 0: **true** und **false** sind Formeln.

Regel 1: eine Aussagenvariable ist eine Formel.

Regel 2: ist F eine Formel, dann ist auch $\neg F$ eine Formel.

Regel 3: sind F und G Formeln, dann sind

a. $(F \wedge G)$
b. $(F \vee G)$
c. $(F \rightarrow G)$

ebenfalls Formeln.

Regel 4: Ein Ausdruck ist nur dann eine Formel, wenn er durch Anwendung der obenstehenden Regeln konstruiert werden kann.

Kapitel II – Grundlagen; Logik

- Syntax der Aussagenlogik: Formationsregeln
  - Es lassen sich also durch Anwendung von logischen Operatoren (Konnektoren) auf Formeln neue Formeln bilden.
  - Ein Operator kombiniert einen oder mehrere Operanden zu einem komplexeren Ausdruck.
  - Monadische/unäre Operatoren haben ein Argument (z.B. $\neg F$), dyadische/binäre Operatoren haben zwei Argumente (z.B. $F \vee G$).

Kapitel II – Grundlagen; Logik

- Formeln sind:

$((p \wedge q) \rightarrow r)$

$((p \wedge (\mathbf{false} \vee q)) \rightarrow \neg r)$

$((\neg p \vee \neg q) \rightarrow (q \wedge p))$

- Keine Formeln sind:

$\wedge (q \rightarrow r))$

$p \wedge (p \wedge q))$

$(\wedge q \rightarrow (q \wedge p))$

Kapitel II – Grundlagen; Logik

- Syntax der Aussagenlogik: Formationsregeln

Regel 0: **true** und **false** sind Formeln.

Regel 1: eine Aussagenvariable ist eine Formel.

Regel 2: ist F eine Formel, dann ist auch $\neg F$ eine Formel.

Regel 3: sind F und G Formeln, dann sind

a. $(F \wedge G)$
b. $(F \vee G)$
c. $(F \rightarrow G)$

ebenfalls Formeln.

Regel 4: Ein Ausdruck ist nur dann eine Formel, wenn er durch Anwendung der obenstehenden Regeln konstruiert werden kann.

Kapitel II – Grundlagen; Logik

- Formeln sind:

$((p \wedge q) \rightarrow r)$

$((p \wedge (\mathbf{false} \vee q)) \rightarrow \neg r)$

$((\neg p \vee \neg q) \rightarrow (q \wedge p))$

- Keine Formeln sind:

$\wedge (q \rightarrow r))$

$p \wedge (p \wedge q))$

$(\wedge q \rightarrow (q \wedge p))$ $\quad \underbrace{(\wedge}$ $\quad (\wedge$

Kapitel II – Grundlagen; Logik

- Formeln sind:

$((p \wedge q) \rightarrow r)$

$((p \wedge (\mathbf{false} \vee q)) \rightarrow \neg r)$

$((\neg p \vee \neg q) \rightarrow (q \wedge p))$

- Keine Formeln sind:

$\wedge (q \rightarrow r))$

$p \wedge (p \wedge q))$

$(\wedge q \rightarrow (q \wedge p))$ $\quad \underbrace{(\wedge}$ $\quad (\wedge$

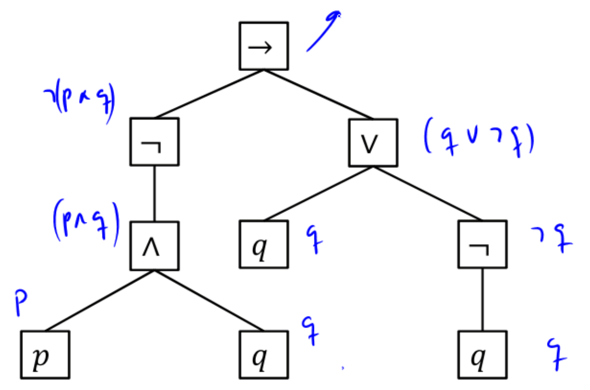
Kapitel II – Grundlagen; Logik

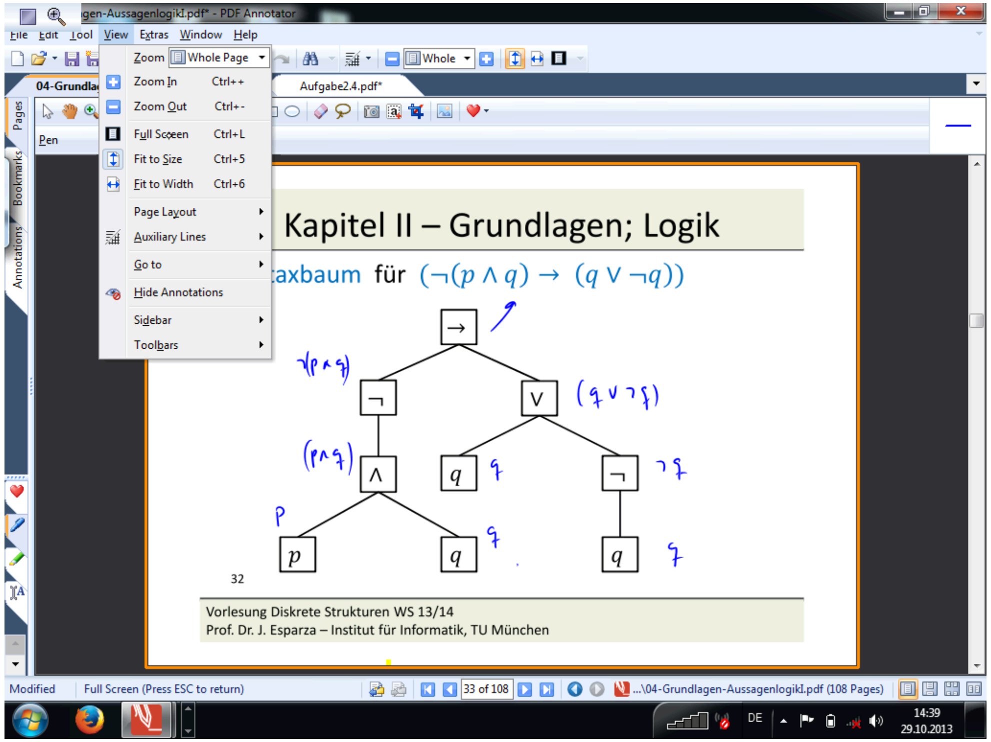
- Ableitungsbeispiel

(1)	p	Regel 1
(2)	q	Regel 1
(3)	$(p \wedge q)$	Regel 3a, (1), (2)
(4)	$\neg(p \wedge q)$	Regel 2, (3)
(5)	$\neg q$	Regel 2, (2)
(6)	$(q \vee \neg q)$	Regel 3b, (2), (5)
(7)	$(\neg(p \wedge q) \rightarrow (q \vee \neg q))$	Regel 3c, (4), (7)

Kapitel II – Grundlagen; Logik

- Syntaxbaum für $(\neg(p \wedge q) \rightarrow (q \vee \neg q))$



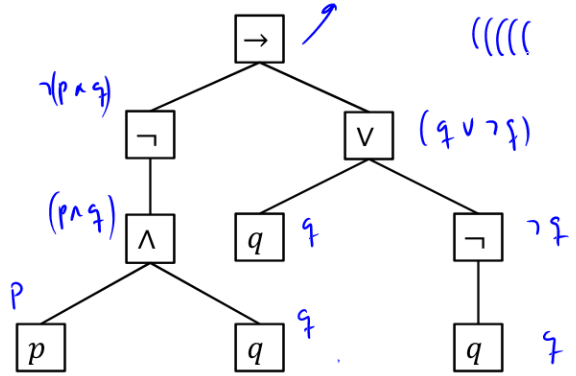


Kapitel II – Grundlagen; Logik

- Bindungsregeln zur Vereinfachung von Klammer-ausdrücken
  - $\neg$ bindet stärker als $\wedge$
  - $\wedge$ bindet stärker als $\vee$
  - $\vee$ bindet stärker als $\rightarrow$
- Beispiele:
  - $\neg p \wedge q$ steht für $(\neg p \wedge q)$, nicht für $\neg (p \wedge q)$
  - $p \wedge q \rightarrow r$ steht für $((p \wedge q) \rightarrow r)$, nicht für $(p \wedge (q \rightarrow r))$
  - $p \wedge q \vee r \wedge s$ steht für $((p \wedge q) \vee (r \wedge s))$
- In den Folien:
  - Weder $p \wedge q \vee r \wedge s$ noch $((p \wedge q) \vee (r \wedge s))$,
  - sondern $(p \wedge q) \vee (r \wedge s)$

Kapitel II – Grundlagen; Logik

- Syntaxbaum für $(\neg(p \wedge q) \rightarrow (q \vee \neg q))$



Kapitel II – Grundlagen; Logik

- Bindungsregeln zur Vereinfachung von Klammer-ausdrücken
  - $\neg$ bindet stärker als $\wedge$
  - $\wedge$ bindet stärker als $\vee$
  - $\vee$ bindet stärker als $\rightarrow$
- Beispiele:
  - $\neg p \wedge q$ steht für $(\neg p \wedge q)$, nicht für $\neg(p \wedge q)$
  - $p \wedge q \rightarrow r$ steht für $((p \wedge q) \rightarrow r)$, nicht für $(p \wedge (q \rightarrow r))$
  - $p \wedge q \vee r \wedge s$ steht für $((p \wedge q) \vee (r \wedge s))$
- In den Folien:

  Weder $p \wedge q \vee r \wedge s$ noch $((p \wedge q) \vee (r \wedge s))$,

  sondern $(p \wedge q) \vee (r \wedge s)$

Kapitel II – Grundlagen; Logik

- Semantik der Aussagenlogik.
  - Was ist die „Bedeutung“ einer Formel? $p \vee q$.
    - Die Formel $F = (\neg p \wedge q)$ kann in vier „Welten“ ausgewertet werden
      - Welt 1: p und q sind beide wahr. F ist falsch.
      - Welt 2: p ist wahr, q ist falsch. F ist falsch.
      - Welt 3: p ist falsch, q ist wahr. F ist wahr
      - Welt 4: p ist falsch, q ist falsch. F ist falsch.

Kapitel II – Grundlagen; Logik

- Semantik der Aussagenlogik.
  - Die Bedeutung einer Formel ist die Funktion, die jede mögliche „Welt“ dem Wahrheitswert der Formel in dieser Welt zuordnet:
    - 1 (die Formel ist wahr), oder
    - 0 (die Formel ist falsch).

  Man nennt {1, 0} die Booleschen Werte.

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.
  - Eine Belegung („eine Welt“) ist eine Funktion von einer Menge von Aussagevariablen in die Menge {0,1} der Wahrheitswerte.
  - Eine Belegung $\beta\colon V \to \{0,1\}$ passt zu einer Formel F, wenn alle Variablen, die in F vorkommen, zu V gehören.
  - Beispiel: die Funktionen
    - $p \mapsto 0,\ \ q \mapsto 1$
    - $p \mapsto 1\,,\ \ q \mapsto 1,\ \ r \mapsto 1$

    sind passende Belegungen der Formel $\neg p \wedge q$.

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.
  - Die Semantik einer Formel F ist eine Funktion $[F]$, die jede Belegung β, die zu F passt, („jede Welt“) einem Wahrheitswert $[F](\beta)$ zuordnet.

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.

  [true]

  - Die Funktion $[F]$ ist folgendermaßen definiert in Abhängigkeit von F:

  (1) Semantik der Formeln **true** und **false**:

  F = **true**. Für alle Belegungen β gilt: $[\mathbf{true}](\beta) = 1$

  F = **false**. Für alle Belegungen β gilt: $[\mathbf{false}](\beta) = 0$

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.

  (2) Semantik der Negation (NICHT) und der Konjunktion (UND): $\neg G$

  $F = \neg G$ für eine Formel G. Für jede Belegung β, die zu F passt:

  $$[F](\beta) = \begin{cases} 0 & \text{falls } [G](\beta) = 1 \\ 1 & \text{falls } [G](\beta) = 0 \end{cases}$$

  $F = (G \wedge H)$ für Formeln G und H. Für jede Belegung β, die zu F passt:

  $$[F](\beta) = \begin{cases} 1 & \text{falls } [F](\beta) = 1 \text{ und } [G](\beta) = 1 \\ 0 & \text{falls } [F](\beta) = 0 \text{ oder } [G](\beta) = 0 \end{cases}$$

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.

  (3) Semantik der Disjunktion (ODER):

  $F = (G \vee H)$ für Formeln G und H. Für jede Belegung β , die zu F passt, gilt:

$$[F](\beta) = \begin{cases} 1 & \text{falls } [F](\beta) = 1 \text{ oder } [G](\beta) = 1 \\ 0 & \text{falls } [F](\beta) = 0 \text{ und } [G](\beta) = 0 \end{cases}$$

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.

  (4) Semantik der Implikation (WENN…DANN):

  $F = (G \rightarrow H)$ für Formeln G und H. Für jede Belegung β, die zu F passt :

$$[F](\beta) = \begin{cases} 1 & \text{falls } [F](\beta) = 0 \text{ oder } [G](\beta) = 1 \\ 0 & \text{falls } [F](\beta) = 1 \text{ und } [G](\beta) = 0 \end{cases}$$

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.
  - Wahrheitstabelle:

F	G	$F \rightarrow G$
0	0	1
0	1	1
1	0	0
1	1	1

  - Semantik in Worten: $F \rightarrow G$ ist falsch genau dann wenn F wahr ist und G falsch ist. Andernfalls ist $F \rightarrow G$ wahr.
  - In der Implikation $F \rightarrow G$ ist F die Hypothese und G die Konklusion.

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.

  Die formale Semantik der Implikation ist unterschiedlich von dem Gebrauch von „wenn … dann“ in der Alltagssprache.

  - $p \rightarrow q$ sagt nicht, dass p eine Ursache für q ist
    - Pinguine schwimmen → Pferde wiehern ist wahr in unserer Welt.
  - $p \rightarrow q$ sagt nichts darüber, ob p wahr oder falsch ist
    - Frau Merkel wurde bestochen → Frau Merkel gehört hinter Gitter ist wahr in unserer Welt.

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.
  - Wahrheitstabelle:

F	G	$F \to G$
0	0	1
0	1	1
1	0	0
1	1	1

  - Semantik in Worten: $F \to G$ ist falsch genau dann wenn F wahr ist und G falsch ist. Andernfalls ist $F \to G$ wahr.
  - In der Implikation $F \to G$ ist F die Hypothese und G die Konklusion.

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.

  Die formale Semantik der Implikation ist unterschiedlich von dem Gebrauch von „wenn … dann" in der Alltagssprache.

  - $p \to q$ sagt nicht, dass p eine Ursache für q ist
    - Pinguine schwimmen → Pferde wiehern ist wahr in unserer Welt.
  - $p \to q$ sagt nichts darüber, ob p wahr oder falsch ist
    - Frau Merkel wurde bestochen → Frau Merkel gehört hinter Gitter ist wahr in unserer Welt.
  - Wenn in einer Welt p falsch ist, dann ist in dieser Welt $p \to q$ wahr, unabhängig davon, was q sagt !!
    - Pinguine fliegen → Pferde sprechen ist wahr in unserer Welt.

Kapitel II – Grundlagen; Logik

- Formale Semantik der Aussagenlogik.
  - $p \to q$ sagt nichts darüber, ob $\neg p \to \neg q$ wahr ist
    - In der Alltagssprache:

      aus „Wenn du brav bist, dann bekommst Du ein Bonbon" folgt implizit „Wenn Du nicht brav bist, dann bekommst Du kein Bonbon"

Kapitel II – Grundlagen; Logik

- Weitere Operatoren: Bikonditional
  – Der binäre Operator "$\leftrightarrow$" entspricht dem sprachlichen Konstrukt "genau dann, wenn ".
  – Wahrheitstabelle:

G	H	$G \leftrightarrow H$
0	0	1
0	1	0
1	0	0
1	1	1

Kapitel II – Grundlagen; Logik

- Aufgabe: Sind diese Aussagen wahr oder falsch in unserer Welt?
  - Diese Vorlesung wird niemals enden → die Sonne wird morgen aufgehen
  - Dienstag ist ein Tag der Woche → Obama ist ein Pinguin.
  - 1+1=6 → Merkel ist Kanzlerin
  - Der Mond ist aus grünem Käse → Frankreich hat einen König

55

Kapitel II – Grundlagen; Logik

- Wahrheitstabellen
  - Sei F eine Formel, und sei V_F die Menge der Variablen, die in F vorkommen.
  - Eine Belegung, die zu F passt, ist minimal für F wenn $V_F = V$
  - Sei β eine Belegung, die zu F passt, und sei β_F die Projektion von β auf V_F. Dann ist β_F ist eine minimale Belegung und es gilt: $[F](\beta) = [F](\beta_F)$.
  - Das heisst: die Funktion $[F]$ wird von den minimalen Belegungen eindeutig bestimmt.
  - Die Wahrheitstabelle von F enthält als Zeilen die minimalen Belegungen von F und die entsprechenden Werte von $[F]$.

56