Einführung in die Informatik I

Winter 2005/2006

6 Logik

Versionsdatum: 10. November 2005

6 Logik ...

Motivation: Ziele dieses Abschnittes

- sprach- und kulturunabhängige Regeln zur Bewertung von Aussagen und Formeln
- Verknüpfung von Aussagen, Schlußfolgerungen, Vereinfachung von Formeln
- Umgang mit formalen Elementen der Logik

Herausforderungen:

- Abstraktion von gewohnten umgangssprachlichen Aussagen und Bewertungen
- Relativ starkes formales Element

Literatur: In [Claus] finden sich verstreut einige Begriffe

- [Ernst] Kap. 4.2 (Aussagenlogik)
- [Wiki]: `www.wikipedia.de` mit den Begriffen Logik, Aussagen- und Prädikatenlogik

Motivation:

Einsatz der Logik u.a. für

- ❍ mathematische Beweisführung
- ❍ Hardwaretechnische Realisierung logischer Funktionen mit Schaltungselementen
- ❍ Formulierung, Umwandlung, Vereinfachungen und Schlußfolgerung von Aussagen z.B. in der Programmierung

Teilgebiete, die hier von Interesse sind

- ❍ ***Aussagenlogik***: Bewertung von Aussagen
  - ❑ Aussagen sind mathematisch oder umgangssprachlich
  - ❑ jede Aussage ist entweder ***wahr*** oder ***falsch*** (Digitalrechner!)
- ❍ ***Prädikatenlogik***: Erweiterung der Aussagenlogik
  - ❑ Untersuchung der Struktur einer Aussage
  - ❑ Hinzunahme von ***Variablen*** und ***Wertebereiche***

Zum Logik-Begriff

- ❍ In der Philosophie, Mathematik und Informatik angesiedelt
- ❍ Vielzahl von Teilgebieten neben der Aussagen- und Prädikatenlogik
- ❍ Charakteristisch: Umformung einer Aussage/Formel ändert Wahrheitswertes nicht

Beispiel:

Mathematik: Äquivalenzumformungen wie: $5 \cdot x = 20 \Leftrightarrow x = 4$

Äquivalenzzeichen „⇔“ symbolisiert die Gleichwertigkeit beider Seiten

Umgangssprache: Aussagen wie

„Heute regnet es“ oder „Das Auto ist rot“

oder ***Folgerungen***

„Wenn es regnet, (dann) ist die Straße nass“

Beispiel: (Forts.)

Aber: Nicht alle umgangssprachlichen Sätze sind Aussagen

- „Dieser Satz ist falsch" ist nicht bewertbar
- „Der Kaffee schmeckt gut" ist nicht allgemein bewertbar

Grundregel

Für die Informatik ist es zwingend, dass Aussagen und Formeln

1. unabhängig von Sprache, Kultur etc. eindeutig interpretiert/bewertet werden
2. vom/im Rechner interpretier- und verarbeitbar sind

   $\Rightarrow$ eindeutige Notation (ähnliche Anforderungen wie bei formalen Sprachen)

6.1 Aussagenlogik

Grundlagen der Aussagenlogik

- Eine ***Elementaraussage*** A ist entweder wahr (1) oder falsch (0)

  Alternative Bezeichnungen: 'F/T' (false, true), 'F/W' (falsch,wahr) 'L/H' (low,high)
- Elementaraussagen können zu ***Formeln*** f verknüpft werden

Beispiele: Elementaraussagen

A_1: München ist 781 km von Hamburg entfernt

A_2: 9 ist durch 3 teilbar

A_3: Alle Autos sind grün

- A_2 ist offensichtlich wahr, A_1 muss zunächst geprüft werden
- A_3 ist eine (offensichtlich falsche) All-Aussage, deren Struktur Teil des Abschnitts ***Prädikatenlogik*** ist. Im Sinne der Aussagenlogik ist A_3 wie die anderen Beispiele jedoch eine Elementaraussage

6.1.1 Junktoren

Allgemein

Junktor: Operator zur Verknüpfung von Aussagen und Formeln zu neuen Formeln

Elementaraussagen beinhalten keine Junktoren

Negation

A	$\neg A$
1	0
0	1

- Umdrehung des Wahrheitsgehaltes einer Aussage A (bzw. Formel f)
- Notation: $\neg A$ (auch $\overline{A}$)
- A und $\neg A$ können nicht gleichzeitig wahr sein
- **Beispiele:**
  Negation der Aussagen der vorherigen Folie
  $\neg A_1$: München ist nicht 781 km von Hamburg entfernt
  $\neg A_2$: 9 ist nicht durch 3 teilbar
  $\neg A_3$: Nicht alle Autos sind grün

6.1.1 Junktoren ...

Konjunktion

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

- „Und“ Verknüpfung zweier Aussagen A, B (analog für Formeln)
- Notation: $f = A \wedge B$ (auch $A \cdot B$ oder AB)
- Formel f ist wahr, wenn A und B wahr sind

Beispiele:

A: 9 ist durch 3 teilbar

B: 9 ist eine Quadratzahl

$f_1 = A \wedge B$: 9 ist durch 3 teilbar und 9 ist eine Quadratzahl

$f_2 = A \wedge \neg B$: 9 ist durch 3 teilbar und 9 ist keine Quadratzahl

$f_3 = \neg A \wedge B$: 9 ist nicht durch 3 teilbar und 9 ist eine Quadratzahl

$f_4 = \neg A \wedge \neg B$: 9 ist nicht durch 3 teilbar und 9 ist keine Quadratzahl

Nur f_1 ist wahr, weil sowohl A als auch B wahr sind

Disjunktion

- „Oder“ Verknüpfung zweier Aussagen A, B (analog für Formeln)
- Notation: $f = A \lor B$ (auch $A + B$)
- Aussage f ist wahr, wenn A oder B wahr ist

A	B	$A \lor B$
1	1	1
1	0	1
0	1	1
0	0	0

Beispiele:

A: 9 ist durch 3 teilbar

B: 9 ist eine Quadratzahl

$f_5 = A \lor B$: 9 ist durch 3 teilbar oder 9 ist eine Quadratzahl

$f_6 = A \lor \neg B$: 9 ist durch 3 teilbar oder 9 ist keine Quadratzahl

$f_7 = \neg A \lor B$: 9 ist nicht durch 3 teilbar oder 9 ist eine Quadratzahl

$f_8 = \neg A \lor \neg B$: 9 ist nicht durch 3 teilbar oder 9 ist keine Quadratzahl

Nur f_8 ist falsch, da hier weder A noch B wahr sind

Implikation

- Folgerung einer Aussage B aus einer Aussage A (analog für Formeln)
- Notation: $f = (A \rightarrow B)$
- Wenn man aus einer wahren Aussage A schließen kann, dass B ebenfalls wahr ist, spricht man von Implikation
- inhaltlich gleichbedeutende Formeln

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

$$(A \rightarrow B) \Leftrightarrow \neg A \lor B \quad \text{und} \quad (A \rightarrow B) \Leftrightarrow (\neg B \rightarrow \neg A)$$

Beispiele:

- „Wenn es regnet, wird die Straße nass“ ist gleichbedeutend mit „Wenn die Straße NICHT nass ist, dann regnet es NICHT“

  Eine falsche Schlußfolgerung wäre „Wenn es nicht regnet, dann ist die Straße nicht nass“
- Wenn Person x ein Auto der Marke BMW hat, hat x ein Auto
- n ist teilbar durch 6, also ist n teilbar durch 3

6.1.1 Junktoren ...

Äquivalenz

- Gleichwertige Aussagen A, B (analog für Formeln)
- Notation: $A \leftrightarrow B$
- gleichwertige Formel :

$$(A \leftrightarrow B) \Leftrightarrow ((A \rightarrow B) \wedge (B \rightarrow A))$$

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Beispiele:

- n ist genau dann durch sechs teilbar, wenn n durch 2 und 3 teilbar ist
- Heute ist genau dann Dienstag, wenn morgen Mittwoch ist

6.1.2 BNF der Aussagenlogik und Bindungsregeln

BNF der Aussagenlogik

Startsymbol: <Formel>

BNF:

```
<Formel>       ::= <Impl_Formel>
                 | <Formel> ↔ <Impl_Formel>
<Impl_Formel>  ::= <Disj_Formel>
                 | <Impl_Formel> → <Disj_Formel>
<Disj_Formel>  ::= <Konj_Formel>
                 | <Disj_Formel> ∨ <Konj_Formel>
<Konj_Formel>  ::= <Term>
                 | <Konj_Formel> ∧ <Term>
<Term>         ::= ¬<Term>
                 | ( <Formel> )
                 | 0 | 1 | <Bezeichner>
```

Bezeichner stellt hierbei die Symbole für Elementaraussagen dar

6.1.2 BNF der Aussagenlogik und Bindungsregeln ...

Bindungsregeln

Problem: Aussagenlogische Formeln werden durch viele Klammern unübersichtlich lang

Bindungsregeln: BNF legt Priorität in der Anwendung fest

1. Die Negation ($\neg$) bindet am Stärksten, dann
2. die Konjunkion ($\wedge$), dann
3. die Disjunktion ($\vee$), dann
4. die Implikation ($\longrightarrow$) und schließlich
5. die Äquivalenz ($\longleftrightarrow$)

Linksbindung: Gleichwertige binäre Operatoren werden linksgeklammert (BNF!)

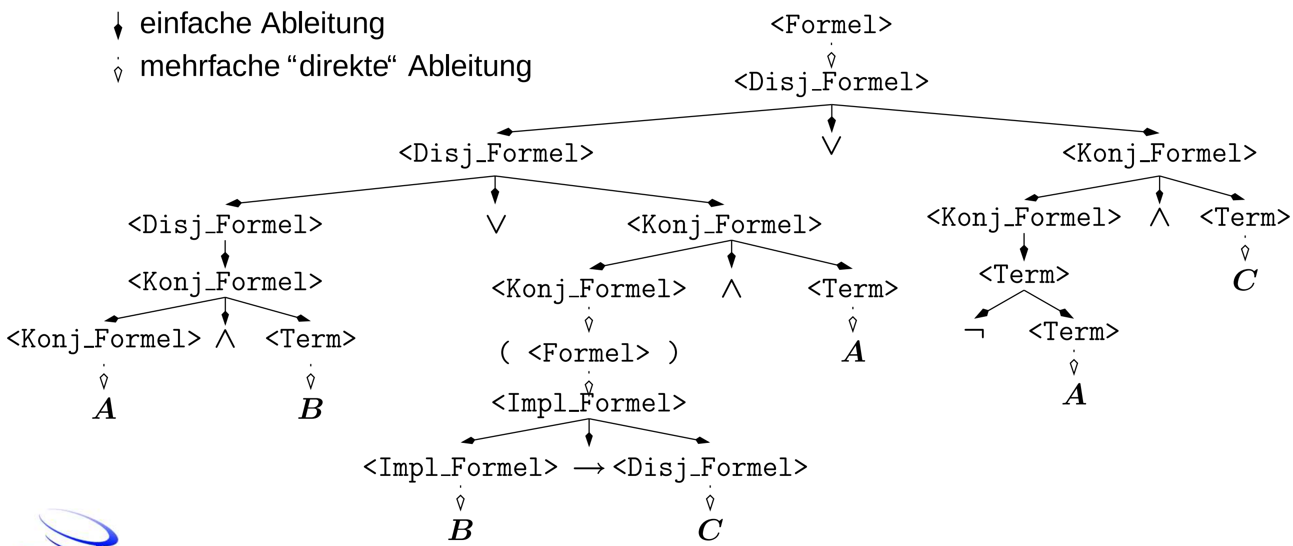
6.1.2 BNF der Aussagenlogik und Bindungsregeln ...

Beispiel: BNF und Bindungsregeln

Gegeben: Formel $f = ((A \wedge B) \vee ((B \rightarrow C) \wedge A)) \vee ((\neg A) \wedge C)$

Vereinfachte Formel: $f = A \wedge B \vee (B \rightarrow C) \wedge A \vee \neg A \wedge C$

Ableitungsbaum:



6.1.3 Umformungsregeln

Ziel: Regeln zur Umformung einer Formel in gleichwertige Formeln zur ***Vereinfachung***

Regeln aufbauend auf Aussagen A, B, C (analog für Formeln)

Regelname	**Gleichwertige Formeln**
Doppelte Negation	$\neg\neg A \Leftrightarrow A$
Idempotenz	$A \vee A \Leftrightarrow A$ und $A \wedge A \Leftrightarrow A$
Kommutativität	$A \vee B \Leftrightarrow B \vee A$, $A \wedge B \Leftrightarrow B \wedge A$
	und $A \leftrightarrow B \Leftrightarrow B \leftrightarrow A$
Assoziativität	$(A \vee B) \vee C \Leftrightarrow A \vee (B \vee C)$ und
	$(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
Distributivität	$(A \wedge B) \vee C \Leftrightarrow (A \vee C) \wedge (B \vee C)$ und
	$(A \vee B) \wedge C \Leftrightarrow (A \wedge C) \vee (B \wedge C)$
DeMorgan	$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$ und $\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$

6.1.3 Umformungsregeln ...

Regelname	**Gleichwertige Formeln**
Tautologie	$\neg A \vee A \Leftrightarrow 1$
Widerspruch	$\neg A \wedge A \Leftrightarrow 0$
Oder-Vereinfachung	$A \vee 1 \Leftrightarrow 1, A \vee 0 \Leftrightarrow A$ und $A \vee (A \wedge B) \Leftrightarrow A$
Und-Vereinfachung	$A \wedge 1 \Leftrightarrow A, A \wedge 0 \Leftrightarrow 0$ und $A \wedge (A \vee B) \Leftrightarrow A$

Beispiel zur Anwendung des De Morgan'schen Gesetzes

A:	n ist durch 3 teilbar
B:	n ist durch 2 teilbar
$f = A \wedge B$:	n ist durch 3 und durch 2 teilbar
$\neg f = \neg(A \wedge B)$:	n ist nicht (gleichermaßen) durch 2 und durch 3 teilbar
$\neg f = \neg A \vee \neg B$:	n ist entweder nicht durch 2 oder nicht durch 3 teilbar

6.1.3 Umformungsregeln ...

Beispiel: Vereinfachung aussagenlogischer Formeln

Gegeben: Formel mit Verknüpfung der drei Aussagen A, B, C

Gesucht: Möglichst einfache, gleichwertige Formel

$$
\begin{array}{ll}
(\neg A \wedge B \wedge \neg C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge B \wedge C) & \\
\Leftrightarrow (\neg A \wedge B \wedge \neg C) \vee ((A \wedge B) \wedge \neg C) \vee ((A \wedge B) \wedge C) & \text{Assoziativität} \\
\Leftrightarrow (\neg A \wedge B \wedge \neg C) \vee ((A \wedge B) \wedge (\neg C \vee C)) & \text{Distrib.-Ges.} \\
\Leftrightarrow (\neg A \wedge B \wedge \neg C) \vee ((A \wedge B) \wedge 1) & \text{Tautologie} \\
\Leftrightarrow (\neg A \wedge B \wedge \neg C) \vee (A \wedge B) & \text{Und-Vereinf.} \\
\Leftrightarrow B \wedge ((\neg A \wedge \neg C) \vee A) & \text{Distrib.-Ges.} \\
\Leftrightarrow B \wedge ((\neg A \vee A) \wedge (\neg C \vee A)) & \text{Distrib.-Ges.} \\
\Leftrightarrow B \wedge (1 \wedge (\neg C \vee A)) & \text{Tautologie} \\
\Leftrightarrow B \wedge (\neg C \vee A) & \text{Und-Vereinf.}
\end{array}
$$

6.1.4 Normalformen

Gegeben: Eine beliebige aussagenlogische Formel f

Ziel: ***Normierung*** (= „Vereinheitlichung“) von f

$\Rightarrow$ Verbesserung der Vergleichbarkeit und der Lesbarkeit der Formel

Normalformen: Die beiden wichtigsten Normalformen sind

1. ***Konjunktive Normalform (KNF)***: Konjunktion von Disjunktionen („Verundung von Oderausdrücken“)
   Beispiel:
   $$(A \vee B) \wedge (\neg A \vee B) \wedge (A \vee \neg B)$$
2. ***Disjunktive Normalform (DNF)***: Disjunktionen von Konjunktion („Veroderung von Undausdrücken“)
   Beispiel:
   $$(A \wedge B) \vee (\neg A \wedge B) \vee (A \wedge \neg B)$$

6.1.4 Normalformen ...

Beispiel: Ableitung der DNF aus einer Wahrheitstabelle

Ausgangslage: Wahrheitstabelle, die für jede Kombination von Wahrheitswerten der Elementaraussagen den Wahrheitswert von f beschreibt

Ziel: Entsprechende Darstellung von f in DNF

Ansatz: Disjunktion von Konjunktionen der Elementarzustände, für die f 1 (wahr) ist

Konkret für Tabelle rechts

$$\begin{aligned} f \;=\; & (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee \\ & (\neg A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \end{aligned}$$

A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

6.1.4 Normalformen ...

Beispiel: Ableitung der KNF aus einer Wahrheitstabelle

Ausgangslage: Wahrheitstabelle wie bei DNF

Ziel: Entsprechende Darstellung von f in KNF

Ansatz für KNF:

1. Erstellung der DNF für $\neg f$
2. Anwendung des DeMorgan'schen Gesetzes

Konkret:

A	B	C	f	$\neg f$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

$$\begin{aligned} \neg f \;=\; & (\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee \\ & (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge C) \\ \Leftrightarrow f \;=\; & \neg(\neg A \wedge \neg B \wedge \neg C) \wedge \neg(A \wedge \neg B \wedge \neg C) \wedge \\ & \neg(A \wedge \neg B \wedge C) \wedge \neg(A \wedge B \wedge C) \\ \Leftrightarrow f \;=\; & (A \vee B \vee C) \wedge (\neg A \vee B \vee C) \wedge \\ & (\neg A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee \neg C) \end{aligned}$$

6.2 Prädikatenlogik

Allgemein

- Auch ***Quantorenlogik*** genannt
- Erweiterung der Aussagenlogik durch ***Variablen*** und ***Geltungsbereiche***
- Beschreibung von Geltungsbereichen über ***Quantoren***
- Beschreibung von Eigenschaften über ***Prädikats-*** und ***Funktionssysmbole***
- Prädikatenlogik ermöglicht Aussagen wie

$$\left.\begin{array}{l}\text{Alle Metalle leiten Strom}\\ \text{Kupfer ist ein Metall}\end{array}\right\} \longrightarrow \text{Kupfer leitet Strom}$$

6.2 Prädikatenlogik ...

Funktion und Prädikat

Funktionen ordnen ***Tupeln*** von n ***Variablen*** einen ***Funktionswert*** zu

Beispiele wie man sie kennt

$$\sin : \begin{array}{ccc}\mathbb{R} & \longrightarrow & [-1,1]\\ x & \longmapsto & \sin(x)\end{array} \qquad \text{binom}: \begin{array}{ccc}\mathbb{R}^2 & \longrightarrow & \mathbb{R}\\ (x,y) & \longmapsto & x^2+2xy+y^2\end{array}$$

Prädikate sind Funktionen, die immer einen Wahrheitswert als Ergebnis haben

Notation: Prädikate werden häufig in Abhängigkeit der Variablen geschrieben: $P(x)$

Beispiele für ein Prädikat mit zwei Variablen

$$P: \begin{array}{ccc}\mathbb{R}^2 & \longrightarrow & \mathbb{B}=\{0,1\}\\ (x,y) & \longmapsto & \begin{cases}1 & \text{falls } x>y\\ 0 & \text{sonst}\end{cases}\end{array} \quad \text{bzw.} \quad P: \begin{array}{ccc}\mathbb{R}^2 & \longrightarrow & \mathbb{B}=\{0,1\}\\ (x,y) & \longmapsto & (x>y)\end{array}$$

6.2.1 Quantoren

Existenzquantor $\exists$

Ziel: Formulierung der Aussage über die Existenz eines Variablen- oder Tupelwertes in einem Wertebereich mit einer bestimmten Eigenschaft

Notation: $\exists x : P(x)$ oder $\exists x \in W : P(x)$

Sprich: Es existiert ein x (aus der Wertemenge W) für das $P(x)$ gilt

Wahrheitswert: Wahr, wenn für (mind.) ein $x \in W$ $P(x)$ wahr ist

Beispiel: Es gibt natürliche Zahlen (Symbol: $\mathbb{N}$), die durch 2 teilbar sind:

$$\exists x \in \mathbb{N} : (x/2) \in \mathbb{N} \qquad \text{(hier ist also } P(x) = ((x/2) \in \mathbb{N})\text{)}$$

Erweiterung: Formulierung, dass es ***genau einen*** Wert gibt, für das das Prädikat gilt; Zeichen: $\exists_1$ (oder $\exists!$)

Beispiel: Es gibt genau eine natürliche Zahl, deren Quadrat 4 ergibt

$$\exists_1 x \in \mathbb{N} : x^2 = 4 \qquad \text{(gilt das auch für } x \in \mathbb{Z}\text{?)}$$

6.2.1 Quantoren ...

Allquantor $\forall$

Ziel: Formulierung einer Aussage für alle Variablen- oder Tupelwerte in einem Wertebereich mit einer bestimmten Eigenschaft

Notation: $\forall x : P(x)$ oder $\forall x \in W : P(x)$

Sprich: Für alle x (aus der Wertemenge W) gilt $P(x)$

Wahrheitswert: Wahr, wenn für jedes $x \in W$ $P(x)$ wahr ist

Beispiele: 1. Für jede natürliche Zahl x ist auch $x + 1$ eine natürliche Zahl:

$$\forall x \in \mathbb{N} : (x + 1) \in \mathbb{N} \qquad \text{(hier ist also } P(x) = ((x + 1) \in \mathbb{N})\text{)}$$

2. Für jeden Monat m gilt, dass er 30 Tage hat

$$\mathcal{M} = \{\text{Jan, Feb, Mär, Apr, Mai, Jun, Jul, Aug, Sep, Okt, Nov, Dez}\}$$
$$|m| := \text{„Anzahl Tage im Monat } m\text{“} \qquad \text{(Definition)}$$
$$\forall m \in \mathcal{M} : |m| = 30$$

(diese Aussage ist natürlich falsch, da $|\text{Jan}| = 31$)

6.2.1 Quantoren ...

Negation von Quantoren

Frage: Welchen Einfluß hat die logische Negation auf die Quantoren?

Existenzquantor: $\neg(\exists x : P(x)) \Leftrightarrow \forall x : (\neg P(x))$

Allquantor: $\neg(\forall x : P(x)) \Leftrightarrow \exists x : (\neg P(x))$

Beispiel:

Alle Autos sind grün

Verneinung: Nicht alle Autos sind grün $\Leftrightarrow$ Es existiert mindestens ein Auto, das nicht grün ist

Beispiel:

Es gibt eine ganze Zahl n mit der Eigenschaft: n ist durch 3 teilbar

Verneinung: Für alle ganzen Zahlen n gilt: n ist nicht durch 3 teilbar

6.2.1 Quantoren ...

Gesetze

Gesetze der Aussagenlogik gelten weiterhin, z.B.

$$(\forall x \in \mathbb{N} : x^2 \in \mathbb{N}) \wedge (\exists_1 y \in \{2, 3, 4, 5\} : y \text{ ist Primzahl})$$

(Wahrheitswert dieser Konjunktion ist: 0)

Zusätzliche Gesetze:

Quantorwechsel	$\neg(\exists x : P(x)) \Leftrightarrow \forall x : \neg P(x)$
	$\neg(\forall x : P(x)) \Leftrightarrow \exists x : \neg P(x)$
Quantortausch	$\exists x : \exists y : P(x, y) \Leftrightarrow \exists y : \exists x : P(x, y)$
	$\forall x : \forall y : P(x, y) \Leftrightarrow \forall y : \forall x : P(x, y)$
Zusammenfassung	$(\exists x : P(x)) \vee (\exists x : Q(x)) \Leftrightarrow \exists x : (P(x) \vee Q(x))$
	$(\forall x : P(x)) \wedge (\forall x : Q(x)) \Leftrightarrow \forall x : (P(x) \wedge Q(x))$

Beachte: $(\forall x : P(x)) \vee (\forall x : Q(x)) \not\Leftrightarrow \forall x : (P(x) \vee Q(x))$ Warum?

Bemerkung: Es existieren noch weitere Gesetze für die Prädikatenlogik

Bit-Logik

- Stellt ***Operatoren*** bereit, die direkt auf Bit-Ebene arbeiten
  $\Rightarrow$ Effizient auf Digitalrechnern umsetzbar
- Erweiterung der Aussagenlogik im Kontext der Rechnerarchitektur
- Wir verwenden in diesem Abschnitt im Hinblick auf folgende Kapitel die Syntax aus C/C++
- In den Beispielen verwenden wir 8-stellige Binärzahlen, grundsätzlich funktioniert die Bit-Logik selbstverständlich mit beliebig langen Binärzahlen
- Erinnerung: $1 =$ true, $0 =$ false

Bitweises „Und“: $\&$

Verknüpft korrespondierende Bits zweier Binärzahlen mit der Konjunktion

$$\left.\begin{array}{r}00110110\\10010101\end{array}\right\}\&\ 00010100 \quad\Leftrightarrow\quad 00110110\&10010101 = 00010100$$

Bitweises „Oder“: $|$

Verknüpft korrespondierende Bits zweier Binärzahlen mit der Disjunktion

$$\left.\begin{array}{r}00110110\\10010101\end{array}\right\}|\ 10110111 \quad\Leftrightarrow\quad 00110110|10010101 = 10110111$$

Bitweise Negation (Einerkomplement): ~

Invertiert eine Binärzahl bitweise

$$\tilde{}(00110110) = 11001001$$

6.3 Bit-Logik ...

Bitweises ausschließendes „Oder“ (XOR): ^

A	B	$A\hat{}B$
1	1	0
1	0	1
0	1	1
0	0	0

- ausschließendes Oder:

$$A\hat{}B = (A \vee B) \wedge (\neg A \vee \neg B)$$

Verknüpft korrespondierende Bits zweier Binärzahlen mit einem

$$\left.\begin{array}{c} 00110110 \\ 10010101 \end{array}\right\} \hat{}\ 10100011 \quad \Leftrightarrow \quad 00110110\hat{}10010101 = 10100011$$

Bemerkung:

Auf [cplusplus] wird dies „Logisches ausschließendes Oder“ genannt!

6.3 Bit-Logik ...

Bitweiser Linksshift: $<< n$

- verschiebt alle Bits einer Binärzahl um n Stellen nach links
- „neue“ n Stellen rechts werden auf 0 gesetzt

Beispiel:

$00110110 << 2 = 11011000$

Bemerkung:

Vernachlässigt man die Stellen, die nach links „verloren gehen“, ist dies eine höchst effiziente Multiplikation mit 2

Bitweiser Rechtsshift: $>> n$

- verschiebt alle Bits einer Binärzahl um n Stellen nach rechts
- „neue“ n Stellen links werden auf 0 gesetzt

Beispiel:

$00110110 >> 2 = 00001101$

Bemerkung:

Vernachlässigt man die Stellen, die nach rechts „verloren gehen“, ist dies eine höchst effiziente Division durch 2