

Grundlagen der Programmierung (Vorlesung 4)

Ralf Möller, FH-Wedel

- Vorige Vorlesung:
 - Boole'sche Logik,
 - Ersetzbarkeitstheorem, Äquivalenztransformationen
 - Logisches Schließen, Folgerbarkeit
- Inhalt dieser Vorlesung
 - Wiederholung: Algebra und logisches Schließen
 - Anwendungen: Lösen von "Logeleien"
- Lernziele:
 - Vertiefung des vermittelten Stoffes

Boole'sche Algebra: Definition (Huntington)

- Grundmenge M
- Zwei zweistellige Operatoren: φ, ψ
- Zu jedem Operator gibt es in M ein neutrales Element $\{\text{NULL}, \text{EINS}\} \subseteq M$, so daß gilt:
 - $x \varphi \text{NULL} \equiv x$
 - $x \psi \text{EINS} \equiv x$
- Zu jedem Element gibt es eindeutig ein Inverses: \cdot^{-1}
 - Für alle $x \in M$ gilt: $x^{-1} \in M, x \varphi x^{-1} = \text{EINS}, x \psi x^{-1} = \text{NULL}$
- Es gelten weiterhin das Kommutativgesetz und das Distributivgesetz

Boole'sche Algebra: Gesetze

■ Kommutativgesetze

- $x \varphi y \equiv y \varphi x$

- $x \psi y \equiv y \psi x$

■ Distributivgesetze

- $x \varphi (y \psi z) \equiv (x \varphi y) \psi (x \varphi z)$

- $x \psi (y \varphi z) \equiv (x \psi y) \varphi (x \psi z)$

Boolesche Algebra: Ableitung (1)

■ Absorptionsgesetze

- $x \varphi (x \psi y) \equiv x$

- $x \psi (x \varphi y) \equiv x$

■ Idempotenzgesetze

- $(x \varphi x) \equiv x$

- $(x \psi x) \equiv x$

Boolesche Algebra: Ableitung (2)

■ Assoziativgesetze

- $(x \varphi y) \varphi z \equiv x \varphi (y \varphi z)$

- $(x \psi y) \psi z \equiv x \psi (y \psi z)$

■ Involution

- $x^{-1-1} \equiv x$

■ De Morgansche Gesetze

- $(x \psi y)^{-1} \equiv x^{-1} \varphi y^{-1}$

- $(x \varphi y)^{-1} \equiv x^{-1} \psi y^{-1}$

Algebra: Zentrale Einsicht

- Sind die Voraussetzungen erfüllt, folgen bestimmte Gesetze
- Der Urbild- und Bildbereich der Operatoren ist dabei vollkommen unerheblich!

Boole'sche Logik vs. Boole'sche Algebra

■ Die Semantik der Boole'schen Logik ist so definiert, daß mit den entsprechenden Null- und Einselementen die Operatoren eine Boole'sche Algebra darstellen

■ $M = \{0, 1\}$

■ $\varphi = \vee$

■ $\psi = \wedge$

■ $.-1 = \neg$

■ NULL = 0

■ EINS = 1

Eine Logelei: Superman existiert nicht!

Wenn Superman das Böse verhindern kann und will, dann wird er es tun. Wenn Superman das Böse nicht verhindern kann, dann ist er machtlos; wenn er es nicht verhindern will, dann ist er böswillig. Superman verhindert das Böse nicht. Wenn Superman existiert, ist er weder machtlos noch böswillig. Darum existiert Superman nicht.

Folgerbarkeit (Logisches Schließen)

■ Gegeben eine Menge von Formeln F_1, \dots, F_k

■ Ist eine Formel G eine Folgerung?

■ Wiederholung:

Eine Formel G heißt eine *Folgerung* der Formeln F_1, \dots, F_k falls für jede Belegung, die sowohl zu F_1, \dots, F_k als auch zu G passend ist, gilt:

Wenn A Modell von $\{F_1, \dots, F_k\}$ ist, dann ist A auch Modell von G .

Wir schreiben $F_1, \dots, F_k \models G$, falls G eine Folgerung von F_1, \dots, F_k ist.

Gesucht wird ein mechanisches Verfahren

- Wir definieren einen Operator \vdash
- $\{F1, F2, \dots, Fk\} \vdash G$
- Ansprüche an \vdash
- Korrektheit
 - Wenn $\{F1, F2, \dots, Fk\} \vdash G$ dann $\{F1, F2, \dots, Fk\} \models G$
- Vollständigkeit
 - Wenn $\{F1, F2, \dots, Fk\} \models G$ dann $\{F1, F2, \dots, Fk\} \vdash G$

Normalformen

Definition (Normalformen)

Ein *Literal* ist eine atomare Formel oder die Negation einer atomaren Formel. (Im ersten Fall sprechen wir von einem *positiven*, im zweiten Fall von einem *negativen* Literal).

Eine Formel F ist in *konjunktiver Normalform* (**KNF**), falls sie eine Konjunktion von Disjunktionen von Literalen ist:

$$F = \left(\bigwedge_{i=1}^n \left(\bigvee_{j=1}^{m_i} L_{i,j} \right) \right),$$

wobei $L_{i,j} \in \{A_1, A_2, \dots\} \cup \{\neg A_1, \neg A_2, \dots\}$

Eine Formel F ist in *disjunktiver Normalform* (**DNF**), falls sie eine Disjunktion von Konjunktionen von Literalen ist:

$$F = \left(\bigvee_{i=1}^n \left(\bigwedge_{j=1}^{m_i} L_{i,j} \right) \right),$$

wobei $L_{i,j} \in \{A_1, A_2, \dots\} \cup \{\neg A_1, \neg A_2, \dots\}$

Umformungsmethode

Gegeben: eine Formel F .

1. Ersetze in F jedes Vorkommen einer Teilformel der Bauart

$$\begin{aligned} \neg\neg G & \text{ durch } G \\ \neg(G \wedge H) & \text{ durch } (\neg G \vee \neg H) \\ \neg(G \vee H) & \text{ durch } (\neg G \wedge \neg H) \end{aligned}$$

bis keine derartige Teilformel mehr vorkommt.

2. Ersetze jedes Vorkommen einer Teilformel der Bauart

$$\begin{aligned} (F \vee (G \wedge H)) & \text{ durch } ((F \vee G) \wedge (F \vee H)) \\ ((F \wedge G) \vee H) & \text{ durch } ((F \vee H) \wedge (G \vee H)) \end{aligned}$$

bis keine derartige Teilformel mehr vorkommt.

Mengendarstellung

- **Klausel:** Menge von Literalen (Disjunktion).

$\{A, B\}$ stellt $(A \vee B)$ dar.

- **Formel:** Menge von Klauseln (Konjunktion).

$\{\{A, B\}, \{\neg A, B\}\}$ stellt $((A \vee B) \wedge (\neg A \vee B))$ dar.

- **Block:** Menge von Formeln (Disjunktion).

$\{F, G\}$ stellt $(F \vee G)$ dar.

Die leere Klausel ist äquivalent zu \perp .

Die leere Formel ist äquivalent zu \top .

Der leere Block ist äquivalent zu \perp .

Zusammenfassung, Kernpunkte



- "Rechnen" mit Formeln
 - Algebra
 - Transformationsgesetze

- Anwendung
 - Programmtransformation (kommt bald)
 - Lösen von Logeleien

Was kommt beim nächsten Mal?



- Resolutionsverfahren