Computational Logic
Algorithmische Logik

Boolesche Algebra
und
Resolution

Ralf Moeller
Hamburg Univ. of Technology

Boole'sche Algebra

- Äquivalenzen als "Transformationsgesetze"
  - Ersetzbarkeitstheorem
- Zentrale Frage:
  - Ist das alles Zufall?
  - Hängen die Gesetze irgendwie zusammen?
- Beispiel:
  - Nehmen wir an, die Äquivalenzen "Kommutativität" und "Distributivität" wurden bewiesen.
  - Muß man die anderen dann noch beweisen?
    Der Beweis über die Semantik ist aufwendig!

Boole'sche Algebra: Zentrale Idee

- Man nehme Operatoren, deren Semantik eine Funktion über einer Grundmenge M ist
  - Über die Elemente von M wollen wir nichts sagen!
  - Ein mögliches Beispiel ist nun M = {0, 1}
- Man nehme an, daß bezüglich der Operatoren gewisse Gesetze (sog. Axiome) gelten
- Nun zeige man, daß unter bestimmten Voraussetzungen andere Gesetze ebenfalls gelten

Boole'sche Algebra: Definition (Huntington)

- Grundmenge M
- Zwei zweistellige Operatoren: φ, ψ
- Zu jedem Operator gibt es in M ein neutrales Element {NULL, EINS} $\subseteq$ M, so daß gilt:
  - $x \ \varphi \ \mathrm{NULL} \equiv x$
  - $x \ \psi \ \mathrm{EINS} \equiv x$
- Zu jedem Element gibt es eindeutig ein Inverses: $\cdot^{-1}$
  - Für alle $x \in M$ gilt: $x^{-1} \in M$, $x \ \varphi \ x^{-1} = \mathrm{EINS}$, $x \ \psi \ x^{-1} = \mathrm{NULL}$
- 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
  - $(x \psi y)^{-1} \equiv x^{-1} \varphi y^{-1}$
  - $(x \varphi y)^{-1} \equiv x^{-1} \psi y^{-1}$

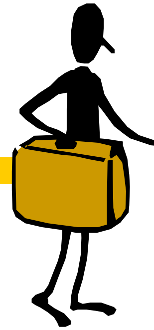
Algebra: Zentrale Einsicht

- 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}
- φ = $\vee$
- ψ = $\wedge$
- $\cdot^{-1}$ = $\neg$
- NULL = 0
- EINS = 1

Zusammenfassung, Kernpunkte



- Aussagenlogik (Boole'sche Logik)
  - Syntax, Formel
  - Semantik, Belegung, Modell
  - Entscheidungsprobleme
- Semantische und Syntaktische Verfahren zur Lösung von Inferenzproblemen
  - Erfüllbarkeit durch Wahrheitstabellen
  - Transformation von Formeln in äquivalente Formeln

Eine Logelei: Supermann existiert nicht!

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

Gesucht wird ein mechanisches Verfahren

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

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:

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

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

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

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

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

Umformungsmethode

Gegeben: eine formel F.

1. Ersetze in F jedes Vorkommen einer Teilformel der Bauart

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

   bis keine derartige Teilformel mehr vorkommt.

2. Ersetze jedes Vorkommen einer Teilformel der Bauart

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

   bis keine derartige Teilformel mehr vorkommt.

Mengendarstellung

- Klausel: Menge von Literalen (Disjunktion).

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

- Formel: Menge von Klauseln (Konjunktion).

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

- Block: Menge von Formeln (Disjunktion).

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

Die leere Klausel ist äquivalent zu $\perp$.

Die leere Formel ist äquivalent zu $\top$.

Der leere Block ist äquivalent zu $\perp$.

Resolvent

Definition Seien K_1, K_2 und R Klauseln. Dann heißt R *Resolvent* von K_1 und K_2, falls es ein Literal L gibt mit $L \in K_1$ und $\overline{L} \in K_2$ und R die Form hat:

$$R = (K_1 - \{L\}) \cup (K_2 - \{\overline{L}\}).$$

Hierbei ist $\overline{L}$ definiert als

$$\overline{L} = \begin{cases} \neg A_i \text{ falls } L = A_i, \\ A_i \text{ falls } L = \neg A_i. \end{cases}$$

Wir stellen diesen Sachverhalt durch folgendes Diagramm dar (Sprechweise: R wird aus K_1, K_2 nach L resolviert).

$$\begin{tikzcd}
K_1 \arrow[dr, no head] & & K_2 \arrow[dl, no head] \\
& R &
\end{tikzcd}$$

Wir vereinbaren ferner, daß die leere Menge, die ebenfalls als Resolvent auftreten kann (falls $K_1 = \{L\}$ und $K_2 = \{\overline{L}\}$ für ein Literal L) mit dem speziellen Symbol $\square$ bezeichnet wird. Dieses Symbol wird verwendet, um eine unerfüllbare Formel zu bezeichnen.

Resolutions–Lemma

Resolutions–Lemma

Sei F eine Formel in **KNF**, dargestellt als Klauselmenge. Ferner sei R ein Resolvent zweier Klauseln K_1 und K_2 in F. Dann sind F und $F \cup \{R\}$ äquivalent.

Beweis:

Sei $\mathcal{A}$ eine zu F (und damit auch zu $F \cup \{R\}$) passende Belegung.

Falls $\mathcal{A} \models F \cup \{R\}$, dann gilt natürlich (erst recht) $\mathcal{A} \models F$.

Sei als umgekehrt angenommen, daß $\mathcal{A} \models F$, d.h. also für alle Klauseln $K \in F$ gilt $\mathcal{A} \models K$. Der Resolvent R habe die Form $R = (K_1 - \{L\}) \cup ((K_2 - \{\overline{L}\})$ mit $K_1, K_2 \in F$ und $L \in K_1$, $\overline{L} \in K_2$.

Fall 1: $\mathcal{A} \models L$.

Dann folgt wegen $\mathcal{A} \models K_2$ und $\mathcal{A} \not\models \overline{L}$, daß $\mathcal{A} \models (K_2 - \{\overline{L}\})$, und damit $\mathcal{A} \models R$.

Fall 2: $\mathcal{A} \not\models L$.

Dann folgt wegen $\mathcal{A} \models K_1$, daß $\mathcal{A} \models (K_1 - \{L\})$ und damit $\mathcal{A} \models R$.

$\boxed{Res(F)}$

Definition

Sei F eine Klauselmenge. Dann ist $Res(F)$ definiert als

$$Res(F) = F \cup \{R \mid R \text{ ist Resolvent zweier Klauseln in } F\}.$$

Außerdem setzen wir:

$$Res^0(F) = F$$
$$Res^{n+1}(F) = Res(Res^n(F)) \text{ für } n \geq 0.$$

und schließlich sei

$$Res^*(F) = \bigcup_{n \geq 0} Res^n(F).$$

Resolutionssatz

- Eine Klauselmenge F ist unerfüllbar genau dann, wenn

$$\square \in Res^*(F)$$

Korrektheit der aussagenlogischen Resolution

$\square \in Res^*(F)$ dann F unerfüllbar

Folgt direkt aus dem Resolutionslemma

Widerlegungsvollständigkeit der aussagenlogischen Resolution

- F unerfüllbar, dann Res*(F) $\vdash \square$
- Induktion über die Anzahl der Variablen
- Anfang: Offensichtlich für n = 1 ($A \wedge \neg A$)
- Annahme: Behauptung gelte für n Variablen
- Zeige: Beh. gilt auch für n+1 Variablen $A_1 \ldots A_{n+1}$
  - Betrachte: KNF(F) mit Fallunterscheidung für A_{n+1} :
    - Wähle A_{n+1} = true:
      - Klauseln mit A_{n+1} positiv entfallen
      - Klauseln mit A_{n+1} negativ haben nur n Variablen
    - Wähle A_{n+1} = false:
      - Analog

Die Supermann-Logelei in Aussagenlogik

- $SBVK \wedge SBVW \rightarrow SBV$
- $\neg SBVK \rightarrow SM$
- $\neg SBVW \rightarrow SB$
- $\neg SBV$
- $SE \rightarrow \neg(SM \vee SB)$

Formelmenge F

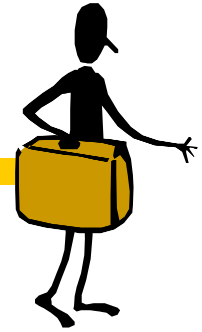
$\models \neg SE$

Folgerung

Lösen des Folgerungsroblems durch Resolution

- Behaupten des Gegenteils durch Negation der Folgerung: ¬ ¬ SE
- Umwandlung der Hypothesenformelmenge in Konjunktion
- Umwandlung der Konjunktion in konjunktive Normalform notiert in Klauselform
- Hinzufügung der negierten Folgerung in KNF notiert in Klauselform
- Zeige: $\square \in Res^*(F \bigcup \{\text{„}Negierte\ Folgerung\text{“}\})$

Zusammenfassung, Kernpunkte



- "Rechnen" mit Formeln
  - Algebra
  - Transformationsgesetze
- Kalkül : Resolution
- Anwendung
  - Programmtransformation (kommt bald)
  - Lösen von Logeleien