Grundlagen der Programmierung

Vorlesung 2 vom 21.10.2004

Sebastian Iwanowski
FH Wedel

Grundlagen der Programmierung

1. Einführung
   - ➔ Grundlegende Eigenschaften von Algorithmen und Programmen
2. Logik
   - Aussagenlogik
   - Prädikatenlogik
3. Programmentwicklung und –verifikation
   - Grundlagen der Programmverifikation
   - Verbundanweisungen
   - Verzweigungen
   - Schleifen
   - Modularisierung
   - Rekursion
4. Entwurf und Analyse von Algorithmen
   - Klassifikation von Algorithmen
   - Programmierung von Algorithmen
   - Bewertung von Algorithmen

Bausteine von Algorithmen

<u>Jeder</u> Algorithmus kann aus folgenden Kontrollstrukturen modular aufgebaut werden:

- **Sequenz**
- **Verzweigung**
- **Schleife**

Zum Lösen komplizierterer Probleme benutzt man zur Vereinfachung noch zwei weitere Techniken:

- **Teilalgorithmen (Unterprogramme)**
- **Rekursion**

➔ **Ein Konzept zur Verifizierung muss nur für diese Kontrollstrukturen erstellt werden.**

➔ **Details in Kapitel 3 dieser Vorlesung**

Beispiel für eine Programmverifikation

Gegeben sei folgender Algorithmus:

```
if (x>0) ∨ ((y+x)≤0)
  then
    z := x • y
  else
    z := x / y
```

Behauptung: **Dieser Algorithmus ist für alle x,y ∈ R ausführbar**

Beweis ?

Frage: **Ist der Algorithmus auch korrekt ?**

Überleitung zum nächsten Kapitel

Wir haben im letzten Beispiel gesehen:

➔ **Das Gebiet der Programmverifikation erfordert einen sicheren Umgang mit formalen logischen Schlüssen**

Es gilt allgemein für alle Aspekte der Programmierung:

! **Grundkenntnisse der Aussagen- und Prädikatenlogik sind unentbehrlich.** !

Daher befassen wir uns zunächst mit dem Thema:

Logik

Grundlagen der Programmierung

1. Einführung
   - Grundlegende Eigenschaften von Algorithmen und Programmen
2. Logik
   - ➔ Aussagenlogik
   - Prädikatenlogik
3. Programmentwicklung und –verifikation
   - Grundlagen der Programmverifikation
   - Verbundanweisungen
   - Verzweigungen
   - Schleifen
   - Modularisierung
   - Rekursion
4. Entwurf und Analyse von Algorithmen
   - Klassifikation von Algorithmen
   - Programmierung von Algorithmen
   - Bewertung von Algorithmen

Aussagen und Wahrheitswerte

Was ist eine Aussage ?

- Eine Aussage ist ein beliebiges Objekt.
- In der Aussagenlogik sind Aussagen unteilbar.
  - Wegen der Unteilbarkeit heißen Aussagen auch *Atome*
  - Da sie wegen der Unteilbarkeit sinnvollerweise mit Buchstaben abgekürzt werden, nennt man Aussagen auch *Literale*

Was ist ein Wahrheitswert ?

- Ein Wahrheitswert ist ein Element aus einer zweielementigen Menge (z.B. dargestellt als {w, f}).

Was macht die Aussagenlogik ?

- Die Aussagenlogik beschäftigt sich mit Funktionen, die jeder Aussage einen Wahrheitswert zuordnen.
  - Solche Funktionen heißen *binäre Funktionen*

Operatoren zwischen Aussagen

Durch Operatoren werden aus alten Aussagen neue Aussagen geschaffen:

Wahrheitswerte für die neuen Aussagen:

p	q	¬p	p ∧ q	p ∨ q	p → q	p ↔ q
w	w	f	w	w	w	w
w	f	f	f	w	f	f
f	w	w	f	w	w	f
f	f	w	f	f	w	w

Einstelliger Operator:

- **Negation (¬)**

Zweistellige Operatoren:

- **Konjunktion (∧)**
- **Disjunktion (∨)**
- **Implikation (→)**
- **Äquivalenz (↔)**

Zusammenhang zwischen den Operatoren

Logische Äquivalenzregeln:

$p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p$

Kontraposition

$p \rightarrow q \Leftrightarrow \neg p \vee q$

Ersetzen der Implikation durch ¬ und ∨

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

Ersetzen der Äquivalenz durch Implikationen

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

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

deMorgansche Regeln

$\neg\neg p \Leftrightarrow p$

Doppelte Negation

Wahrheitswerte für die neuen Aussagen:

p	q	¬p	p ∧ q	p ∨ q	p → q	p ↔ q
w	w	f	w	w	w	w
w	f	f	f	w	f	f
f	w	w	f	w	w	f
f	f	w	f	f	w	w

$p \wedge q \Leftrightarrow q \wedge p$

$p \vee q \Leftrightarrow q \vee p$

Kommutativgesetze

$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$

$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

Distributivgesetze

Zusammenhang zwischen den Operatoren

Logische Schlussregeln:

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

Modus ponens

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

Modus tollens

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

Kettenschluss

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

Indirekter Beweis

Wahrheitswerte für die neuen Aussagen:

p	q	¬p	p ∧ q	p ∨ q	p → q	p ↔ q
w	w	f	w	w	w	w
w	f	f	f	w	f	f
f	w	w	f	w	w	f
f	f	w	f	f	w	w

$p \wedge q \Rightarrow p$ **$p \wedge q \Rightarrow q$**

Logische Einschränkung

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

Logischer Ausschluss

Warum einfach, wenn es auch kompliziert geht ?

Folgende Notationen sind in der Literatur ebenfalls gebräuchlich:

für logische Schlussregeln: $\dfrac{p\ ,\ q}{r}$ *ist dasselbe wie* $p \wedge q \Rightarrow r$

In der Logik unterscheidet man zwischen Schlussfolgerungen in verschiedenen Anwendungen:

$\vdash$, $\models$, $\prec$, $\rightarrow$ *entsprechen der logischen Implikation:* $\Rightarrow$

$=$, $\equiv$, $\prec\succ$, $\leftrightarrow$ *entsprechen der logischen Äquivalenz:* $\Leftrightarrow$

Aussagenlogische Formeln

- **Eine aussagenlogische Formel ist eine Verknüpfung von endlich vielen Literalen mit logischen Operatoren.**

- **Eine Belegung einer Formel ist eine Zuweisung von Wahrheitswerten an die Literale derart, dass dieselben Literale immer denselben Wahrheitswert erhalten.**

  Die Formel als ganze bekommt durch die Belegung ebenfalls einen Wahrheitswert.

Aussagenlogische Formeln

- **Eine Formel heißt erfüllbar, wenn es eine Belegung gibt derart, dass die Formel den Wahrheitswert `w` hat.**

  Eine Formel, in der jeder Literal höchstens einmal vorkommt, ist immer erfüllbar!

  Eine Formel, in der keine Negation vorkommt, ist immer erfüllbar!

  ⇒ **Nur Formeln, die einen Literal mehrfach und mindestens eine Negation enthalten, könnten unerfüllbar sein.**

- **Eine Formel heißt Tautologie oder gültig, wenn sie bei jeder Belegung den Wahrheitswert `w` hat.**

- **Eine Formel heißt widersprüchlich, wenn sie bei keiner Belegung den Wahrheitswert `w` hat.**

Aussagenlogische Formeln

Erfüllbarkeitsproblem (Satisfiability, SAT):

Wie bekommt man heraus, ob eine gegebene Formel erfüllbar ist ?

Beim nächsten Mal: Aussagenlogik, 2. Teil

Boolesche Algebren
Normalformen
Systematische Umformungsverfahren
Vollständigkeit und Korrektheit