Formale Systeme

Logik: Verifikation von Aussagen über Algorithmen

20.12.2012

Ana Sokolova statt Robert Elsässer

Verifikation von Aussagen über Algorithmen

- Hoaresche Logik: Kalkül zum Beweisen von Aussagen über Algorithmen und Programme, Programm-Verifikation [C.A.R. Hoare, 1969]

- Statische Aussagen über Zustände (Werte von Variablen), die der Algorithmus (das Programm) an bestimmten Stellen annehmen kann, z. B.
  ... {pegel < max} pegel := pegel +1;... {0 < i ∧ i < 10} a[i] := 42;...;...

- Aussagen müssen beweisbar für alle Ausführungen des Algorithmus gelten. Im Gegensatz zum dynamischen Testen: Ausführen des Algorithmus für bestimmte Eingaben.

Verifikation von Aussagen über Algorithmen

- Schlussregeln für Anweisungformen erlauben logische Schlüsse über Anweisungen hinweg:

  {pegel+1 ≤ max} pegel := pegel + 1; **{pegel ≤ max}**
  wegen Schlussregel für Zuweisungen

- Verifikation beweist, dass
  - an einer bestimmten Programmstelle eine Aussage über Zustände gilt
  - vor und nach der Ausführung eines Programmstückes eine Invariante gilt
  - ein Algorithmus aus jeder zulässigen Eingabe die geforderte Ausgabe berechnet z.b. {a, b ∈ N} Euklidischer Algorithmus {x = ggT von a, b}
  - eine Schleife terminiert

- Ein Algorithmus und die Aussagen dazu sollen zusammen konstruiert werden.

Vorschau auf Konzepte

- Aussagen charakterisieren Zustände der Ausführung
- Algorithmen in informeller Notation
- Schlussregeln für Anweisungsformen anwenden
- Invariante von Schleifen (und anderen Konstrukten)
- Schlussketten über Anweisungen hinweg verifizieren Aussagen
- Nachweis der Terminierung von Schleifen

Beispiel zur Vorschau: Verifikation des Algorithmus ggT

Vorbedingung: x, y ∈ N, sei G größter gemeinsame Teiler von x und y

Nachbedingung: a = G

Algorithmus mit
a := x; b:= y;
solange a ≠ b wiederhole
falls a > b :

a := a - b;
sonst

b := b - a;

{Aussagen über Variable}:
{INV: G ist ggT von a und b ∧ a>0 ∧ b>0}
{INV ∧ a ≠ b}
{G ist ggT von a und b ∧ a>0 ∧ b>0 ∧ a>b}
→ {G ist ggT von a-b und b ∧ a-b>0 ∧ b>0}

{INV}
{G ist ggT von a und b ∧ a>0 ∧ b>0 ∧ b>a}
→ {G ist ggT von a und b-a ∧ a>0 ∧ b-a>0}

{INV ∧ a=b} → {a = G}

Terminierung

Notation von Algorithmenelementen

Anweisungsform	Notation	Beispiel
Sequenz	Anweisung1; Anweisung2	a := x; b := y
Zuweisung	Variable := Ausdruck	a := x

Alternative	falls Bedingung :
	  Anweisung1
	sonst
	  Anweisung2
		falls a > b :
		  a := a-b
		sonst
		  b := b-a

bedingte Anweisung	falls Bedingung :
	  Anweisung
		falls a > b :
		  a := a-b

Unteralgorithmus	ua()	ggT()

Schleife	solange Bedingung wiederhole
	  Anweisung
		solange a ≠ b w-rhole
		  falls a > b : …

Vor- und Nachbedingung von Anweisungen

{P} A1 {Q} A2 {R}

Vorbedingung von A1

Nachbedingung von A1
Vorbedingung von A2

Nachbedingung von A2

- Zur Verifikation eines Algorithmus muss für jede Anweisung A ein Nachweis gefürt werden
  {P} A {Q}
  Wenn vor der Ausfürung die Anweisung A die Aussage P gilt, dann gilt Q nach der Ausfürung von A, falls A terminiert

- Die Aussagen werden entsprechend der Struktur von A verknüpft
  Für jede Anweisungsform, eine Schlussregel

- Eine Spezifikation liefert eine Vorbedingung und eine Nachbedingung des gesamtes Algorithmus
  {Vorbedingung} Algorithmus {Nachbedingung}

Zuweisungsregel

$$\{P[x/e]\}\ x := e\ \{P\}$$

Substitution - x ist durch e substituiert

Wenn man zeigen will, dass nach der Zuweisung eine Aussage P für x gilt, muss man zeigen, dass vor der Zuweisung dieselbe Aussage P für e gilt!

Sequenzregel

$$\frac{\{P\}\ A1\ \{Q\} \qquad \{Q\}\ A2\ \{R\}}{\{P\}\ A1;A2\ \{R\}}$$

Wenn {P} A1 {Q} und {Q} A2 {R} korrekte Schlüsse sind, dann ist auch {P} A1;A2 {R} ein korrekter Schluss!

Konsequenzregeln

$$\frac{\{P\}\ A\ \{R\} \qquad R \rightarrow Q}{\{P\}\ A\ \{Q\}}$$

Abschwächung der Nachbedingung

$$\frac{P \rightarrow R \qquad \{R\}\ A\ \{Q\}}{\{P\}\ A\ \{Q\}}$$

Verschärfung der Vorbedingung

Regel für Alternative

$$\frac{\{P \wedge B\}\ \ A1\ \{Q\} \qquad \{P \wedge \neg B\}\ A2\ \{Q\}}{\{P\}\ \text{Falls } B\text{: } A1 \text{ sonst } A2\ \{Q\}}$$

Aus der gemeinsamen Vorbedingung P führen beide Zweige auf dieselbe Nachbedingung Q!

Regel für bedingte Anweisung

$$\frac{\{P \wedge B\} \quad A1 \ \{Q\} \qquad \{P \wedge \neg B\} \rightarrow \{Q\}}{\{P\} \text{ Falls } B: A1 \ \{Q\}}$$

Aus der gemeinsamen Vorbedingung P führen die Anweisung und die Implikation auf dieselbe Nachbedingung Q!

Aufrufregel

{P} UA() {Q}

Der Unteralgorithmus UA habe keine Parameter und liefere kein Ergebniss. Seine wirkung auf globale Variable sei spezifiziert durch die Vorbedingung P und die Nachbedingung Q. Dann gilt!

Ohne Parameter und Ergebniss ist diese Regel nur von sehr begrenztem Nutzen

Schleifenregel

$$\frac{\{INV \wedge B\}\ S\ \{INV\}}{\{INV\}\ \text{solange } B \text{ wiederhole } S\ \{INV \wedge \neg B\}}$$

INV ist eine Schleifeninvariante, sie gilt an folgenden Stellen: vor der Schleife, vor und nach jeder Ausfrung von S und nach der Schleife

Terminierung von Schleifen

- Die terminierung einer Schleife solange B wiederhole S
  muss separat nachgewiesen werden
  1. Gib einen ganzzahligen Ausdruck E an über Variablen, die in der Schleife vorkommen, und zeige, dass E bei jeder Iteration durch S verkleinert wird
  2. Zeige dass E nach unten begrenzt ist, z.B. dass $E \geq 0$ eine Konsequenz einer Invariante der Schleife ist.
  es kann auch eine andere Grenze als 0 gewählt werden, E kann auch vergrösert werden und nach oben begrenzt sein!

- Nichtterminierung wird bewiesen, in dem man zeigt,
  1. dass $R \wedge B$ Vor- und Nachbedingung von S ist
  2. dass es eine Eingabe gibt, so dass $R \wedge B$ vor der Schleife gilt
  R kann einen speziellen Zustand charakterisieren in dem die Schleife nicht anhält

- Es gibt Schleifen, für die man nicht entscheiden kann, ob sie für jede Vorbedingung terminieren.

Denksportaufgabe zu Invarianten

- In einem Topf seien s schwarze und w weisse Kugeln und s + w > 0 (s ≥ 0, w ≥ 0)

```
solange mindestens 2 Kugeln im Topf sind
        nimm zwei beliebige Kugeln heraus
        falls sie gleiche Farbe haben:
                wirf beide weg und
                lege eine neue schwarze Kugel in den Topf
        sonst
                lege die weisse Kugel zurück in den Topf und
                wirf die schwarze Kugel weg
```

- Welche Farbe hat die lätzte Kugel?

- Finden Sie Invarianten, die die Frage beantworten!