Vorlesung vom 30.05.06: Totale Korrektheit

Wo sind wir?

- Hoare-Kalkül:
  - Korrektheit und Vollständigkeit des Hoare-Kalküls für *partielle* Korrektheit
  - Generierung von Verifikationsbedingungen
  - *Diese Woche*: Totale Korrektheit
- Jenseits von Hoare: JML und WHY
- Werkzeugunterstützung
- Korrekte Programmentwicklung

Warum totale Korrektheit?

- *Intutiver*: Annahme Termination *implizit* bei Spezifikation
- Siehe *Beispiel*:

$$\vdash \{T\}\texttt{skip}\{T\}$$
$$\vdash \{T \wedge T\}\texttt{skip}\{T\}$$
$$\vdash \{T\}\texttt{while}\ T\ \texttt{do skip od}\{T \wedge \neg T\}$$

Totale Korrektheit

- Totale Korrektheit: $[A]P[B]$ gilt gdw.
  Wenn Zustand $\sigma \models A$, dann:
  - P terminiert
  - Nach Ausführung gilt B
- Formal:

$$[A]c[B] = \forall I.\forall \sigma.(\sigma \models^I A \Longrightarrow \mathcal{C}[\![c]\!]\sigma \neq \bot \wedge \mathcal{C}[\![c]\!]\sigma \models^I B)$$

  - NB: Quantoren — c muß nicht immer terminieren.
  - Für Termination ist Interpretation fest aber beliebig.

Regeln für Totale Korrektheit

- Implikation ✓
- Skip ✓
- Zuweisungsregel ✓!
- Sequenzregel ✓
- Fallunterscheidung ✓
- Schleifenregel ✘

Zuweisungen

$X := A$ Problem: Auswertung von A *muss terminieren*

- *Rekursive* Funktionen
  - Nicht in IMP — aber für *Erweiterungen*!
- *Fehlerfälle*:
  - Division durch 0, Bereichsüberschreitung
  - Werden *nicht* durch $\bot$ modelliert — $\bot$ ist *Nichttermination*
  - Fehlerwerte (e.g. für Fließkommazahlen IEEE 754 und 854: `Inf`, `NaN`)
  - Ausnahmen (e.g. Haskell, Java: Division durch 0 für `Int`)
  - Modulo-Arithmetik (e.g. `Int` $\cong \mathbb{Z}_{2^N}$ mit Wortbreite N)

Schleifenregel

- Idee: *wohlfundierte* Ordnung — hier $\mathbb{N}$
  - Genauer *positive ganze Zahlen*
- Jede Iteration muss *Variante* erniedrigen
- Formal:

$$\frac{\vdash [I \wedge b \wedge E = n]c[I \wedge E < n] \quad \vdash I \wedge b \Longrightarrow 0 \leq E}{\vdash [I]\texttt{while}\ b\ \texttt{do}\ c\ \texttt{od}[I \wedge \neg b]}\ While$$

$$\frac{\vdash [I \wedge b]c[I] \quad \vdash [I \wedge b \wedge E = n]c[E < n] \quad \vdash I \wedge b \Longrightarrow 0 \leq E}{\vdash [I]\texttt{while}\ b\ \texttt{do}\ c\ \texttt{od}[I \wedge \neg b]}\ While$$

wobei n nicht in I, c, b, E auftritt.

- Andere Regeln bleiben

Hoare-Kalkül für totale Korrektheit

$$\frac{}{\vdash [A]\texttt{skip}[A]}\ Skip \qquad \frac{}{\vdash [B[a/X]]X := a[B]}\ Assgn$$

$$\frac{\vdash [A]c_0[C] \quad \vdash [C]c_1[B]}{\vdash [A]c_0; c_1[B]}\ Seq$$

$$\frac{\vdash [A \wedge b]c_0[B] \quad \vdash [A \wedge \neg b]c_1[B]}{\vdash [A]\texttt{if}\ b\ \texttt{then}\ c_0\ \texttt{else}\ c_1\ \texttt{fi}[B]}\ Cond$$

$$\frac{\vdash [I \wedge b]c[I] \quad \vdash [I \wedge b \wedge E = n]c[E < n] \quad \vdash I \wedge b \Longrightarrow 0 \leq E}{\vdash [I]\texttt{while}\ b\ \texttt{do}\ c\ \texttt{od}[I \wedge \neg b]}\ While$$

$$\frac{\vdash (A \Longrightarrow A') \quad \vdash [A']c[B'] \quad \vdash (B' \Longrightarrow B)}{\vdash [A]c[B]}\ Impl$$

Beispiel

- Sei $p \equiv \texttt{while}\ Y \leq R\ \texttt{do}\ R := R - Y;\ Q := Q + 1\ \texttt{od}$
- Zu zeigen: $\vdash [0 < Y] p [true]$
- Schleifenregel
  - Invariante: $I \equiv 0 < Y$
  - Variante: $E \equiv R$

Verifikationsbedingungen für Totale Korrektheit

- Benötigt *Annotation* mit *Varianten*:

$$\begin{aligned} c ::= {} & \texttt{skip} \mid x := a \mid \\ & c; (x := a) \mid c; \{D\} c_0 \\ & \texttt{if}\ b\ \texttt{then}\ c\ \texttt{else}\ c\ \texttt{fi} \mid \texttt{while}\ b\ \texttt{do}\ \{D\}[E] c\ \texttt{od} \end{aligned}$$

$$\begin{aligned} vc([A]\texttt{while}\ b\ \texttt{do}\ \{D\}[E]c\ \texttt{od}[B]) = {} & vc([D \wedge b \wedge E = n] c [D \wedge E < n]) \\ & \cup \{ D \wedge b \Longrightarrow 0 \leq E, \\ & \quad A \Longrightarrow D, \\ & \quad D \wedge \neg b \Longrightarrow B \} \end{aligned}$$

wobei n nicht in A, b, D, c, B, E auftritt

- *Andere* Regeln bleiben.

Verifikationsbedingungen für Totale Korrektheit

vc : **Assn** $\times$ **Com**$_A$ $\times$ **Assn** $\rightarrow \mathbb{P}($**Assn**$)$

$$\begin{aligned} vc([A]\texttt{skip}[B]) &= \{A \Longrightarrow B\} \\ vc([A] X := a [B]) &= \{A \Longrightarrow B\,[a/X]\} \\ vc([A] c; X := a [B]) &= vc([A] c [B\,[a/X]]) \\ vc([A] c; \{D\} c_0 [B]) &= vc([A] c [D]) \cup vc([D] c_0 [B]) \\ vc([A]\texttt{if}\ b\ \texttt{then}\ c_1\ \texttt{else}\ c_2\ \texttt{fi}[B]) &= vc([A \wedge b] c_1 [B]) \\ & \quad \cup vc([A \wedge \neg b] c_2 [B]) \\ vc([A]\texttt{while}\ b\ \texttt{do}\ \{D\}[E]c\ \texttt{od}[B]) &= vc([D \wedge b] c [D]) \\ & \quad \cup vc([D \wedge b \wedge E = n] c [E < n]) \\ & \quad \cup \{ D \wedge b \Longrightarrow 0 \leq E, \\ & \qquad A \Longrightarrow D, \\ & \qquad D \wedge \neg b \Longrightarrow B \} \end{aligned}$$

Beispiel 3: `sum`

$$\begin{aligned} \texttt{sum} &\equiv i := 0; s := 0; \\ & \quad \{J\}\,\texttt{while not}\ (i = n)\ \texttt{do}\ \{I\}[E] i := i + 1; s := s + i\ \texttt{od} \\ pre &\equiv 0 \texttt{ <= } n \\ post &\equiv s = sum(1, n) \\ I &\equiv s = sum(1, i) \\ J &\equiv i = 0\ \texttt{and}\ s = 0 \\ E &\equiv n - i \end{aligned}$$

Aber: Invariante zu schwach.

Beispiel 3a: `sum`

$$\begin{aligned} \texttt{sum} &\equiv i := 0; s := 0; \\ & \quad \{J\}\,\texttt{while}\ i < n\ \texttt{do}\ \{I\}[E] s := s + i; i := i + 1\ \texttt{od} \\ pre &\equiv 0 \texttt{ <= } n \\ post &\equiv s = sum(1, n) \\ J &\equiv i = 0\ \texttt{and}\ s = 0 \\ I &\equiv s = sum(1, (i+1))\ \texttt{and}\ i \texttt{ <= } n \\ E &\equiv n - i \end{aligned}$$

Beispiel 4: `times`

$$\begin{aligned} \texttt{times} &\equiv i := 0; p := 0; \\ & \quad \{J\}\,\texttt{while}\ i < n\ \texttt{do}\ \{I\}[E] p := p + m; i := i + 1\ \texttt{od} \\ pre &\equiv 0 \texttt{ <= } n \\ post &\equiv p = m * n \\ J &\equiv i = 0\ \texttt{and}\ p = 0 \\ I &\equiv p = m * i\ \texttt{and}\ i \texttt{ <= } n \\ E &\equiv n - i \end{aligned}$$

Beispiel 5: `div`

$$\begin{aligned} \texttt{div} &\equiv q := 0; r := x; \\ & \quad \{J\}\,\texttt{while not}\ (r < y)\ \texttt{do}\ \{I\}[E] r := r - y; q := q + 1\ \texttt{od} \\ pre &\equiv 0 \texttt{ <= } x\ \texttt{and}\ 0 < y \\ post &\equiv x = q * y + r\ \texttt{and}\ 0 \texttt{ <= } r\ \texttt{and}\ r < y \\ J &\equiv q = 0\ \texttt{and}\ r = x \\ I &\equiv x = q * y + r\ \texttt{and}\ 0 \texttt{ <= } r\ \texttt{and}\ 0 < y \\ E &\equiv r \end{aligned}$$

Beispiel 6: `euclid`

$$\begin{aligned} \texttt{euclid} &\equiv a := x; b := y; \{J\} \\ & \quad \texttt{while not}\ (a = b)\ \texttt{do} \\ & \qquad \{I\}[E]\ \texttt{if}\ a \texttt{ <= } b\ \texttt{then}\ b := b - a \\ & \qquad\qquad \texttt{else}\ a := a - b \\ & \qquad\qquad \texttt{fi od} \\ pre &\equiv 0 < x\ \texttt{and}\ 0 < y \\ post &\equiv a = \gcd(x, y) \\ J &\equiv x = a\ \texttt{and}\ b = y \\ I &\equiv \gcd(x, y) = \gcd(a, b)\ \texttt{and}\ 0 < a\ \texttt{and}\ 0 < b \\ E &\equiv a + b \end{aligned}$$

Beispiel 7: `pow`

$$
\begin{array}{rcl}
\texttt{pow} & \equiv & \texttt{while not } (y = 0) \texttt{ do} \\
& & \quad \{I_1\}[E_1] \texttt{ while } even(y)) \texttt{ do} \\
& & \qquad\quad \{I_2\}[E_2]\, x := x \texttt{ * } x; y := y/2; \\
& & \qquad\quad \texttt{od;} \\
& & \quad z := z \texttt{ * } x; y := y - 1 \\
& & \quad \texttt{od} \\
pre & \equiv & x = m \texttt{ and } y = n \texttt{ and } z = 1 \\
post & \equiv & z = m^n \\
I_1 & \equiv & m^n = z * x^y \\
I_2 & \equiv & m^n = z * x^y \\
E_1 & \equiv & y(?) \\
E_2 & \equiv & y(?)
\end{array}
$$

Beispiel 8: `isqr`

$$
\begin{array}{rcl}
\texttt{isqr} & \equiv & t := 1; s := t; i := 0; \{J\} \\
& & \texttt{while } s \texttt{ <= } a \texttt{ do } \{I\}[E] \\
& & \ t := t + 2; \\
& & \ s := s + t; \\
& & \ i := i + 1 \texttt{fi od} \\
pre & \equiv & 0 \texttt{ <= } a \\
post & \equiv & i \texttt{ * } i \texttt{ <= } a \texttt{ and } a \texttt{ < } (i + 1) \texttt{ * } (i + 1) \\
J & \equiv & t \texttt{ = } 1 \texttt{ and } s \texttt{ = } 1 \texttt{ and } i \texttt{ = } 0 \\
I & \equiv & ? \\
E & \equiv & a - s(?) \\
Hinweis: & & 1 + 3 + 5 + \ldots + (2n - 1) = n^2
\end{array}
$$

Zusammenfassung

- *Totale* Korrektheit = *Partielle* Korrektheit plus *Termination*

$$[A]c[B] = \{A\}c\{B\} \wedge [A]c[true]$$

- Kalkül für totale Korrektheit: Änderung an *Schleifenregel*
  - `while` erlaubt Nichttermination
  - *Variante* garantiert Termination
- Einfache Verallgemeinerung von *Kalkül* und *Verifikationsbedingungen*
- Nächste Wochen:
  Jenseits des Hoare-Kalküls — die *Java Modelling Language* (JML)