5 Programmverifikation

Spezifizieren - Implementieren - Verifizieren

Prädikatenlogik - While-Programme - Hoare-Kalkül

5.1 Definition Sei (S, Σ) Signatur, V Variablenmenge. Eine **partielle Korrektheitsaussage** über (S, Σ) und V ist eine Zeichenreihe der Form $\{\varphi\}\alpha\{\psi\}$ mit α Programm und φ (**Vorbedingung**), ψ (**Nachbedingung**) Formeln über (S, Σ) und V.

Eine partielle Korrektheitsaussage heißt in einer (S, Σ) Algebra A **gültig**, falls für alle Zustände $z, z' \in \mathcal{Z}(A, V)$ gilt:
$(A \models_z \varphi$ und $z[[\alpha]]_A z') \rightsquigarrow A \models_{z'} \psi$

Schreibweise: $A \models \{\varphi\}\alpha\{\psi\}$

5.2 Bemerkung

Beachte: Es wird somit zugesichert, dass das Programm α, wenn es in einem Zustand, in dem φ gilt, gestartet wird, und wenn es terminiert, einen Zustand in dem ψ gilt, berechnet. Gilt φ im Zustand z, aber terminiert α vom Startzustand z aus nicht (d. h. es gibt gar kein z' mit $z[[\alpha]]_A z'$), so wird *nichts* ausgesagt.

Terminierung kann durch **totalen Korrektheitsaussagen** erfasst werden : $[\boldsymbol{\varphi}]\boldsymbol{\alpha}[\boldsymbol{\psi}]$. Wir behandeln diese hier nicht.
Partielle Korrektheit ist eine logische Eigenschaft, hingegen hängt die Terminierung von Wohlordnungen ab. Siehe Loeckx, Sieber oder Sperschneider, Antoniou. Terminierungsbedingungen werden hauptsächlich in Verbindung mit Schleifen verwendet.

Beispiele (Fort.)

5.3 Beispiel Programm α über Nat von 4.26

$\alpha :: Y := 0;\ Z := 0;$

$$\{Z = Y * Y\}$$

while $\neg Y = X$ **do**

$\quad Z := succ(Z + (Y + Y));\ Y := succ(Y);$

end;

$$\{Y = X \wedge Z = X * X\}$$

α berechnet die Funktion $f(x) = x^2$ in der Variablen Z.

Gültigkeit der partiellen Korrektheitsaussage

$$\{X = X\}\alpha\{Z = X * X\} \text{ in } Nat$$

Abkürzung für Formel, die für jeden Zustand gültig ist: **true**, analog **false** Formel, die für keinen Zustand gültig ist.

Zeige: $Nat \models \{true\}\alpha\{Z = X * X\}$

Beweis: Sei z Zustand, der Variablen in α belegt (genügt $z(X)$!) Es gelte $z[[\alpha]]z'$. Zu zeigen ist $z'(Z) = z'(X)^2$ in Nat.

- $z[[Y := 0;\ Z := 0]]z_1$, dann ist $z_1(Y) = 0, z_1(Z) = 0$ und $z_1(Z) = z_1(Y)^2$.
- Ist $z(X) = 0$, so fertig.

Beispiele (Fort.)

- Sonst
  $z_1[[Z := succ(Z + (Y + Y));\ Y := succ(Y)]]z_2$
  mit
  $z_2(Y) = z_1(Y) + 1$
  $z_2(Z) = z_1(Z) + 2z_1(Y) + 1 = z_1(Y)^2 + 2z_1(Y) + 1$
  $= (z_1(Y) + 1)^2 = z_2(Y)^2$
- Ist $z(X) = 1$, so fertig.
- Induktion nach $z(X)$ liefert Behauptung, da $z_i(X) = z(X)$ und beim Austreten aus der While-Schleife $z_n(Y) = z'(Y) = z(X)$.

Für die Programme β und γ über N gilt entsprechend:

$$Nat \models \{true\}\beta\{Z = X + Y\}$$

bzw.

$$Nat \models \{true\}\gamma\{Z = X * Y\}$$

Sind dies auch partielle Korrektheitsaussagen über der Signatur von N und sind sie gültig in N?

Gibt es eine Möglichkeit den Nachweis von Korrektheitsaussagen systematisch zu führen?

Kalkül zur Ableitung partieller Korrektheitsaussagen

5.4 Definition Der Kalkül von Hoare

φ, ψ, ξ seien Formeln über einer Signatur (S, Σ) und Variablenmenge V, X Variablenbezeichner in V, t ein Term vom selben Typ, B eine boolesche Formel und α, β Programme über $(S, \Sigma), V$.

Regeln des hoareschen Kalküls (HC)

Regeln für das leere Programm:

$$\frac{(\varphi \rightarrow \psi)}{\{\varphi\}\varepsilon\{\psi\}}$$

Regeln für Zuweisungen:

$$\frac{\varphi \rightarrow [\psi]\{X/t\}}{\{\varphi\}X := t;\ \{\psi\}}$$

Regeln für Testanweisungen:

$$\frac{\{(\varphi \wedge B)\}\alpha\{\psi\}, \{(\varphi \wedge \neg B)\}\beta\{\psi\}}{\{\varphi\}\underline{\textbf{if}}\ B\ \underline{\textbf{then}}\ \alpha\ \underline{\textbf{else}}\ \beta\ \underline{\textbf{end}};\ \{\psi\}}$$

Kalkül zur Ableitung partieller Korrektheitsaussagen (2)

Regeln für Schleifen:

$$\frac{(\varphi \to \xi), \{(\xi \wedge B)\}\alpha\{\xi\}, ((\xi \wedge \neg B) \to \psi)}{\{\varphi\}\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{end}};\ \{\psi\}}$$

ξ wird **Schleifeninvariante** genannt.

Regeln für Anweisungsfolgen:

$$\frac{\{\varphi\}\alpha\{\xi\}, \{\xi\}\beta\{\psi\}}{\{\varphi\}\alpha\beta\{\psi\}}$$

Für eine Algebra A ist $\mathbf{HC}(A)$ die Erweiterung von HC um die Menge aller in A gültigen prädikatenlogischen Formeln über die Signatur von A als Axiome. Schreibweise $\underset{HC(A)}{\vdash} \{\varphi\}\alpha\{\psi\}$.

5.5 Bemerkung Objekte sind hier PL-Formeln und partielle Korrektheitsaussagen.

Ziel ist es gültige Korrektheitsaussagen in einer Algebra A abzuleiten. Dies wird durch Ableitungen in HC(A) realisiert. Alles was in HC(A) abgeleitet werden kann, sollte in A gültig sein.

Zuweisungsregel: Andere Formen z. B. als Axiom

$$\overline{\{[\varphi]\{X/t\}\}X := t;\ \{\varphi\}}$$

klar aus Substitutionslemma

$A \models_z [\varphi]\{X/t\}$ gdw $A \models_{z(X/a)} \varphi$

Kalkül zur Ableitung partieller Korrektheitsaussagen (3)

Abschwächung der Vor- und Nachbedingungen

$$\frac{\varphi \to \psi, \{\psi\}\alpha\{\xi\}, \xi \to \eta}{\{\varphi\}\alpha\{\eta\}}$$

Dies kann simuliert werden mit HC(A) (über Regel für das leere Programm und Regel für Anweisungsfolgen mit $\alpha\varepsilon = \varepsilon\alpha = \alpha$ für jedes Programm α).

Bei der Anwendung der Schleifenregel ist eine geeignete Invariante zu finden. Gibt es stets eine Formel, die als Invariante verwendet werden kann ?

Bei der Anwendung der Anweisungsfolgenregel ist eine geeignete „Zwischenformel“ ξ zu finden.

Schnittstelle zur Datenstruktur (Theorie von A): Über Zuweisungsregel und Regel für das leere Programm.
Schwierigkeiten bei der Programmverifikation: Der Nachweis von Eigenschaften der Datenstruktur ist für viele Datenstrukturen nicht effektiv durchzuführen (z.B. für Nat).

Notation für Ableitungen im hoareschen Kalkül

5.6 Definition Programme mit Kommentaren

- Kommentar: { PL-Formel }
  Erweiterung der Syntax von Programmen:

- $\{\varphi\}X := t;\ \{\psi\}$

- $\{\varphi\}$**if** B **then** $\{(\varphi \wedge B)\}\alpha\{\psi\}$ **else** $\{(\varphi \wedge \neg B)\}\beta\{\psi\}$ **end**; $\{\psi\}$

- $\{\varphi\}\{\xi\}$**while** B **do** $\{(\xi \wedge B)\}\alpha\{\xi\}$ **end**; $\{(\xi \wedge \neg B)\}\{\psi\}$

- $\{\varphi\}\alpha\{\xi\}\beta\{\psi\}$

Ein Programm ist **syntaktisch korrekt kommentiert**, wenn die Kommentare im Programm diese Regeln erfüllen. Für die Herleitbarkeit im Kalkül $HC(A)$ benötigt man nur noch die **Beweisverpflichtungen** als Theoreme in A nachzuweisen.
Gezeigt werden müssen:
$A \models (\varphi \rightarrow \psi)$,
wenn $\{\varphi\}\{\psi\}$ oder $\{\varphi\}\varepsilon\{\psi\}$ im kommentierten Programmtext vorkommt, bzw.
$A \models (\varphi \rightarrow [\psi]\{X/t\})$,
wenn $\{\varphi\}X := t;\ \{\psi\}$ im kommentierten Programmtext vorkommt.

Notation für Ableitungen im hoareschen Kalkül (Forts.)

Vollständig kommentierte Programme:

Zwischen zwei Anweisungen stets Kommentar.

$\{\varphi\}$
A_1
$\{\ldots\}$
$\{\ldots\}$
A_2
$\{\ldots\}$
$\{\ldots\}$
A_3
$\{\ldots\}$
$\{\ldots\}$
$\ldots$
$\{\ldots\}$
$\{\ldots\}$
A_n
$\{\psi\}$

Hilfsmittel beim Nachweis von $\underset{HC(A)}{\vdash} \{\varphi\}A_1 \ldots A_n\{\psi\}$

Werden bei einem vollständig kommentierten Programm alle Beweisverpflichtungen als gültig in A nachgewiesen, so gilt
$\underset{HC(A)}{\vdash} \{\varphi\}A_1 \ldots A_n\{\psi\}$.

Beweis !

5.7 Beispiel

Fort. Beispiel 4.26 kommentiertes Programm α für $f(x) = x^2$ über Nat. Spezifikation Pre::{true} Post::$\{Z = (X * X)\}$.

$$
\begin{array}{ll}
 & \{\text{true}\} \\
 & Y := 0; \\
 & \{Y = 0\} \\
 & Z := 0; \\
\varphi :: & \{Y = 0 \wedge Z = 0\} \\
\xi :: & \{(Y < X \vee Y = X) \wedge Z = (Y * Y)\} \\
 & \underline{\textbf{while}}\ \neg Y = X\ \underline{\textbf{do}} \\
\xi \wedge B :: & \{(Y < X \vee Y = X) \wedge Z = Y * Y \wedge \neg Y = X\} \\
 & Z := succ(Z + (Y + Y)); \\
 & \{(Y < X \wedge Z = (succ(Y) * succ(Y))\} \\
 & Y := succ(Y); \\
\xi :: & \{(Y < X \vee Y = X) \wedge Z = (Y * Y)\} \\
 & \underline{\textbf{end}}; \\
\xi \wedge \neg B :: & \{((Y < X \vee Y = X) \wedge Z = (Y * Y)) \wedge \\
 & \neg\neg Y = X\} \\
 & \{Z = (X * X)\}
\end{array}
$$

Syntaktisch korrekt kommentiert, vollständig.

Beispiel (Forts.)

Die Ableitbarkeit in HC(Nat) folgt nun aus den Nachweis der Beweisverpflichtungen:

$$
\begin{aligned}
(1)Nat &\models \text{true} \rightarrow 0 = 0 \\
(2)Nat &\models Y = 0 \rightarrow (Y = 0 \wedge 0 = 0) \\
(3)Nat &\models (Y = 0 \wedge Z = 0) \rightarrow ((Y < X \vee Y = X) \wedge \\
&\quad Z = (Y * Y)) \\
(4)Nat &\models (((Y < X \vee Y = X) \wedge Z = Y * Y) \wedge \neg Y = X) \\
&\quad \rightarrow (Y < X \wedge succ(Z + (Y + Y)) = \\
&\quad succ(Y) * succ(Y)) \\
(5)Nat &\models (Y < X \wedge Z = succ(Y) * succ(Y)) \rightarrow \\
&\quad (succ(Y) < X \vee succ(Y) = X) \wedge Z = \\
&\quad succ(Y) * succ(Y) \\
(6)Nat &\models (((Y < X \vee Y = X) \wedge Z = Y * Y) \wedge \\
&\quad \neg\neg Y = X) \rightarrow Z = X * X
\end{aligned}
$$

Korrektheit des hoareschen Kalküls

5.8 Satz

Ist die partielle Korrektheitsaussage $\{\varphi\}\alpha\{\psi\}$ in HC(A) ableitbar, so ist $\{\varphi\}\alpha\{\psi\}$ in A gültig.

D. h.

$$\underset{HC(A)}{\vdash} \{\varphi\}\alpha\{\psi\} \rightsquigarrow A \models \{\varphi\}\alpha\{\psi\}$$

Beweis: Strukturelle Induktion (oder Induktion über Länge der Ableitung in HC(A)).

Regel für das leere Programm:
Vor: $A \models (\varphi \rightarrow \psi)$

z. Z. $A \models \{\varphi\}\varepsilon\{\psi\}$.

Seien z, z' Zustände mit $A \models_z \varphi$ und $z[[\varepsilon]]_A z'$. Nach Definition der Semantik gilt $z = z'$, also wegen $A \models_{z'} \varphi$ und $A \models_{z'} (\varphi \rightarrow \psi)$ auch $A \models_{z'} \psi$.

Regel für Zuweisung:
Vor: $A \models \varphi \rightarrow [\psi]\{X/t\}$

z. Z. $A \models \{\varphi\}X := t;\ \{\psi\}$.

Seien z, z' Zustände mit $A \models_z \varphi$ und $z[[X := t]]_A z'$. Nach Definition der Semantik gilt $z' = z(X/a)$ mit $a = val_{A,z}(t)$. Wegen $A \models_z [\psi]\{X/t\}$ folgt $A \models_{z(X/a)} \psi$ aus Substitutionslemma.

Korrektheit des hoareschen Kalküls (2)

Regel für Testanweisung:
Vor: $A \models \{(\varphi \wedge B)\}\beta\{\psi\}$, $A \models \{(\varphi \wedge \neg B)\}\gamma\{\psi\}$

z. Z. $A \models \{\varphi\}$**if** B **then** β **else** γ **end**; $\{\psi\}$.

Seien z, z' Zustände mit $A \models_z \varphi$ und $z[[$**if** B **then** β **else** γ **end**; $]]_A z'$.

Im Fall $A \models_z B$ folgt $A \models_z (\varphi \wedge B)$ und $z[[\beta]]_A z'$. Aus $A \models \{(\varphi \wedge B)\}\beta\{\psi\}$ folgt $A \models_{z'} \psi$.
Analog Fall $A \models_z \neg B$.

Regel für Schleifen:
Vor: $A \models (\varphi \rightarrow \xi)$, $A \models \{(\xi \wedge B)\}\alpha\{\xi\}$ und
$A \models ((\xi \wedge \neg B) \rightarrow \psi)$

z. Z. $A \models \{\varphi\}$**while** B **do** β **end**; $\{\psi\}$

Sei $A \models_z \varphi$ und $z[[$**while** B **do** β **end**$]]_A z'$. Nach Definition von $[[\cdot]]_A$ gibt es $t \in \mathbb{N}$ und Zustände $z_0, \ldots, z_t$ mit $z = z_0$, $A \models_{z_i} B$ und $z_i[[\beta]]_A z_{i+1}$, $0 \leq i < t$, $A \models_{z_t} \neg B$ und $z_t = z'$.

Korrektheit des hoareschen Kalküls (3)

Daraus ergibt sich:

$A \models_{z_0} \varphi \quad (z = z_0), A \models_{z_0} \xi \quad$ (da $A \models \varphi \to \xi$)
$A \models_{z_0} (\xi \wedge B)$ (B gilt in z_0), $A \models_{z_1} \xi$ (Invarianz von ξ)
$A \models_{z_1} (\xi \wedge B) \cdots$
$\cdots$
$A \models_{z_{t-1}} (\xi \wedge B)$ (B gilt in z_{t-1}) $A \models_{z_t} \xi$ (Invarianz von ξ)
$A \models_{z_t} (\xi \wedge \neg B)$ B gilt nicht mehr
$A \models_{z_t} \psi \quad$ (da $A \models (\xi \wedge \neg B) \to \psi$)
$A \models_{z'} \psi$ Beh.

Regel für Anweisungsfolgen:
Vor: $A \models \{\varphi\}\alpha\{\xi\}$, $A \models \{\xi\}\beta\{\psi\}$

z. Z. $A \models \{\varphi\}\alpha\beta\{\psi\}$

Sei $A \models_z \varphi$ und $z[[\alpha\beta]]_A z'$ für Zustände z, z'.

Nach Definition von $[[\cdot]]_A$ gibt es Zustand z'' mit $z[[\alpha]]_A z''$ und $z''[[\beta]]_A z'$. Nach Vor. $A \models_{z''} \xi$ und auch $A \models_{z'} \psi$.

Abgeleitete Regeln - Vereinfachungen

5.9 Bemerkung Der Hoarsche Kalkül bleibt korrekt wenn er um die Regel für Zuweisungsfolgen erweitert wird:

$$\frac{(\varphi \rightarrow [\cdots[\psi]\{X_n/t_n\}\cdots]\{X_1/t_1\})}{\{\varphi\}X_1 := t_1;\ \cdots\ ;\ X_n := t_n;\ \{\psi\}} \quad \mathsf{Typ}(X_i) = \mathsf{Typ}(t_i)$$

Dies folgt, da die Regel durch Anwenden der Regeln

$$\frac{(\varphi \rightarrow [\cdots[\psi]\{X_n/t_n\}\cdots]\{X_1/t_1\})}{\{\varphi\}X_1 := t_1;\ \{[\cdots[\psi]\{X_n/t_n\}\cdots]\{X_2/t_2\}\}}$$

$$\frac{([\cdots[\psi]\{X_n/t_n\}\cdots]\{X_i/t_i\} \rightarrow [\cdots[\psi]\{X_n/t_n\}\cdots]}{\{[\cdots[\psi]\{X_n/t_n\}\cdots]\{X_i/t_i\}\}X_i := t_i;\ \{X_i/t_i\}}$$
$$\overline{\{[\cdots[\psi]\{X_n/t_n\}\cdots]\{X_{i+1}/t_i\}}$$

für $(i = 2, \ldots, n)$ und iteriertes Anwenden der Anweisungsfolgenregel simulieren lässt.

Als Kommentar in den Programmtext übertrage die Regel als

$$\begin{array}{l}\{\varphi\}\\ X_1 := t_1;\\ \cdots\\ X_n := t_n;\\ \{\psi\}\end{array}$$

Als Beweisverpflichtung muss gezeigt werden

$A \models (\varphi \rightarrow [\cdots[\psi]\{X_n/t_n\}\cdots]\{X_1/t_1\})$

Beachte dabei die Reihenfolge der Substitutionen.

Beispiele

5.10 Beispiel Programm, das den Wert der Variablen X, Y tauscht.
X, Y, X', Y', Z seien Variablen vom gleichen Typ.

$$\{(X = X' \wedge Y = Y')\}$$
$$Z := X;\ X := Y;\ Y := Z;$$
$$\{(Y = X' \wedge X = Y')\}$$

z. Z.

$A \models (X = X' \wedge Y = Y') \rightarrow$
$[[[(Y = X' \wedge X = Y')]\{Y/Z\}]\{X/Y\}]\{Z/X\} =$
$[[(Z = X' \wedge X = Y')]\{X/Y\}]\{Z/X\} =$
$[(Z = X' \wedge Y = Y')]\{Z/X\} =$
$(X = X' \wedge Y = Y')$

d. h. z. Z.:

$$A \models (X = X' \wedge Y = Y') \rightarrow (X = X' \wedge Y = Y')$$

was richtig ist.

Problemspezifikation:

- Festlegung der Signatur (S, Σ).
- Festlegung der Algebra A.
- Festlegung der Vor- und Nachbedingung φ, ψ.
- Festlegung weiterer Hilfsinformation.

Finde Programm α mit $A \models \{\varphi\}\alpha\{\psi\}$.

Beispiel: GGT-Berechnung

5.11 Beispiel Algebra Nat mit Signatur-Erweiterung

$- : nat \times nat \to nat$ mit $n -_{Nat} m = \begin{cases} 0 & m > n \\ n - m & \text{sonst} \end{cases}$

(Übung: Schreibe while-Programm über Signatur von Nat dafür).

$X \mid Y \equiv \exists V V * X = Y, \quad X \leq Y \equiv (X = Y \vee X < Y)$

als Abkürzungen, dann gilt in Nat

$\mathsf{GGT}(X, Y) = Z \equiv$
$Z \mid X \wedge Z \mid Y \wedge \forall V((V \mid X \wedge V \mid Y) \to V \mid Z)$

φ:: $\{X > 0, Y > 0\}$
$A := X;\ B := Y;$
$\{\xi :: \{X > 0, Y > 0, \mathrm{GGT}(X, Y) = \mathrm{GGT}(A, B)\}\}$
while $A \neq B$ **do** $\{\xi, A \neq B\}$
if $B < A$
then $\{\xi, A \neq B, B < A\}$
$A := A - B;$ $\{\xi\}$
else $\{\xi, A \neq B, \neg B < A\}$
$B := B - A;$ $\{\xi\}$
end; $\{\xi\}$
end; $\{\xi, A = B\}$

ψ:: $\{A = \mathsf{GGT}(X, Y)\}$

Beispiel GGT-Berechnung (Forts.)

Die nachzuweisenden Beweisverpflichtungen in Nat sind:

- $(X > 0 \wedge Y > 0) \rightarrow [[\xi]\{B/Y\}]\{A/X\}$
  d. h.
  $(X > 0 \wedge Y > 0) \rightarrow (X > 0 \wedge Y > 0 \wedge \mathsf{GGT}(X, Y) = \mathsf{GGT}(X, Y))$
- $(\xi \wedge A \neq B \wedge B < A) \rightarrow [\xi]\{A/A - B\}$
  d. h.
  $(X > 0 \wedge Y > 0 \wedge \mathsf{GGT}(X, Y) = \mathsf{GGT}(A, B) \wedge B < A) \rightarrow (X > 0 \wedge Y > 0 \wedge \mathsf{GGT}(X, Y) = \mathsf{GGT}(A - B, B))$
- $(\xi \wedge A \neq B \wedge \neg B < A) \rightarrow [\xi]\{B/B - A\}$
  d. h.
  $(X > 0 \wedge Y > 0 \wedge \mathsf{GGT}(X, Y) = \mathsf{GGT}(A, B) \wedge A < B) \rightarrow (X > 0 \wedge Y > 0 \wedge \mathsf{GGT}(X, Y) = \mathsf{GGT}(A, B - A))$
- $(\xi \wedge A = B) \rightarrow A = \mathsf{GGT}(X, Y)$
  d. h.
  $X > 0 \wedge Y > 0 \wedge \mathsf{GGT}(A, B) = \mathsf{GGT}(X, Y) \wedge A = B) \rightarrow A = \mathsf{GGT}(X, Y)$

Diese sind „leicht“ über Nat als gültig nachzuweisen.

Beispiel: Problemspezifikation

5.12 Beispiel Finde Programm über den natürlichen Zahlen mit Division und Modulofunktion, das zu zwei Zahlen X, Y die Zahl X^Y berechnet.

$\mathrm{div} : nat \times nat \to nat$ ganzzahlige Division
$\mathrm{mod} : nat \times nat \to nat$ Rest modulo ...

(Hilfsfunktionen: Übung: Schreibe Programme über Nat für diese Funktionen)

Vorbedingung: true $(0 = 0)$ **Eingabevariable**: $A, B : nat$
Nachbedingung: $Z = A^B$ **Ausgabevariable**: $Z : nat$

(Hierbei steht $Z = A^B$ als Abkürzung für PL-Formel über Signatur von Nat, die A^B beschreibt: A^B steht für $A * \cdots * A$ B-mal $(B = 0$, so 1, d.h. auch $0^0 = 1))$

Existenz einer solchen Formel wird vorausgesetzt!

Formeln um Eigenschaften zu beschreiben, insbesondere Folgen mit bestimmten Eigenschaften $\rightsquigarrow$ Logik.

Verwendet werden im Programm nur

$$Y \mathrm{mod}\ 2 \text{ und } Y \mathrm{div}\ 2$$

Programm

Programm α mit $nat \models \{0 = 0\}\alpha\{Z = A^B\}$:

Verwendete Idee: $X^{2k} = (X^2)^k$
$X^{2k+1} = X^{2k} * X$

$\{0 = 0\}$
$X := A;\ Y := B;\ Z := 1;$
$\{\xi :: \{X^Y * Z = A^B\}$
while $\neg Y = 0$ **do**
$\{ \qquad \}$
if $Y \bmod 2 = 0$
then
$\{ \qquad \}$
$Y := Y \operatorname{div} 2;\ X := X * X;$
$\{ \qquad \}$
else
$\{ \qquad \}$
$Y := Y - 1;\ Z := Z * X;$
$\{ \qquad \}$
end;
$\{ \qquad \}$
end; $\{X^Y * Z = A^B, Y = 0\}$
$\{Z = A^B\}$

Beweisverpflichtungen

Verbleibende Beweisverpflichtungen: in Nat.

- $\models 0 = 0 \to A^B * 1 = A^B$
- $\models (\neg Y = 0 \wedge Y \bmod 2 = 0 \wedge X^Y * Z = A^B) \to (X * X)^{Y \operatorname{div} 2} * Z = A^B$
- $\models (\neg Y = 0 \wedge \neg Y \bmod 2 = 0 \wedge X^Y * Z = A^B) \to X^{Y-1} * Z * X = A^B$
- $\models (X^Y * Z = A^B \wedge Y = 0) \to Z = A^B$

Die Gültigkeit dieser Formeln folgt aus „Idee“ und aus Bedeutung von A^B.

Frage: Kalkül von Hoare ist korrekt. Ist er auch für jede Signatur und Algebra A vollständig? D. h. gilt

aus $\models_A \{\varphi\}\alpha\{\psi\}$ folgt stets $\underset{HC(A)}{\vdash} \{\varphi\}\alpha\{\psi\}$ für φ, ψ PL-Formeln und α While-Programme.

Nein nur falls A **ausdrucksstark** (oder expressiv).

Frage:

1. Gegeben α, ψ. Gibt es stets eine Formel φ mit $A \models \{\varphi\}\alpha\{\psi\}$?
   $\rightsquigarrow$ „Schwächste Vorbedingung“
2. Gegeben φ, α. Gibt es stets eine Formel ψ mit $A \models \{\varphi\}\alpha\{\psi\}$?
   $\rightsquigarrow$ „Stärkste Nachbedingung“