Axiomatische Semantik von WHILE

Insbesondere: ...Korrektheit und Vollständigkeit der axiomatischen Semantik

Erinnerung:

- *Hoare-Tripel* (syntaktische Sicht) bzw. *Korrektheitsformeln* (semantische Sicht) der Form

$$\{p\}\ \pi\ \{q\} \quad \text{bzw.} \quad [p]\ \pi\ [q]$$

- Gültigkeit einer Korrektheitsformel im Sinne
  - *partieller* Korrektheit
  - *totaler* Korrektheit

Definition partieller Korrektheit

Sei $\pi \in$ **Prg** ein WHILE-Programm:

Ein Hoaresche Zusicherung $\{p\}\ \pi\ \{q\}$ heißt

- *gültig* (*im Sinne der partiellen Korrektheit*) oder kurz (*partiell*) *korrekt* gdw. für jeden Anfangszustand σ gilt: ist die Vorbedingung p in σ erfüllt **und** terminiert die zugehörige Berechnung von π angesetzt auf σ regulär in einem Endzustand σ', **dann** ist auch die Nachbedingung q in σ' erfüllt.

Definition totaler Korrektheit

Sei $\pi \in$ **Prg** ein WHILE-Programm:

Ein Hoaresche Zusicherung $[p]\ \pi\ [q]$ heißt

- *gültig* (*im Sinne der totalen Korrektheit*) oder kurz (*total*) *korrekt* gdw. für jeden Anfangszustand σ gilt: ist die Vorbedingung p in σ erfüllt, **dann** terminiert die zugehörige Berechnung von π angesetzt auf σ regulär mit einem Endzustand σ' **und** die Nachbedingung q ist in σ' erfüllt.

Intuitiv

"Totale Korrektheit = Partielle Korrektheit + Terminierung"

Partielle und totale Korrektheit

- Die Zustandsmenge

$$Ch(p) =_{df} \{\sigma \in \Sigma \mid [\![\, p \,]\!]_B(\sigma) = \mathtt{tt}\}$$

heißt *Charakterisierung von* $p \in$ **Bexp**.

- *Semantik von Korrektheitsformeln:*
  Eine Korrektheitsformel $\{p\}\ \pi\ \{q\}$ heißt
  - *partiell korrekt* (in Zeichen: $\models_{pk} \{p\}\ \pi\ \{q\}$), falls $[\![\, \pi \,]\!](Ch(p)) \subseteq Ch(q)$
  - *total korrekt* (in Zeichen: $\models_{tk} \{p\}\ \pi\ \{q\}$), falls $\{p\}\ \pi\ \{q\}$ partiell korrekt ist und $Def([\![\, \pi \,]\!]) \supseteq Ch(p)$ gilt. Dabei bezeichnet $Def([\![\, \pi \,]\!])$ die Menge aller Zustände, für die π regulär terminiert.

  Konvention: $[\![\, \pi \,]\!](Ch(p)) =_{df} \{[\![\, \pi \,]\!](\sigma) \mid \sigma \in Ch(p)\}$

Erinnerung

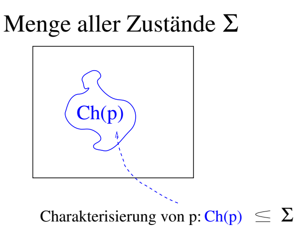
...an einige Sprechweisen:

Ein (deterministisches) Programm π

- angesetzt auf einen Anfangszustand σ *terminiert regulär* gdw. π nach endlich vielen Schritten in einem Zustand $\sigma' \in \Sigma$ endet.
- angesetzt auf einen Anfangszustand σ *terminiert irregulär* gdw. π nach endlich vielen Schritten zur Konfiguration $undef$ führt.
- Ein Programm π heißt *divergent* gdw. π terminiert für keinen Anfangszustand regulär.

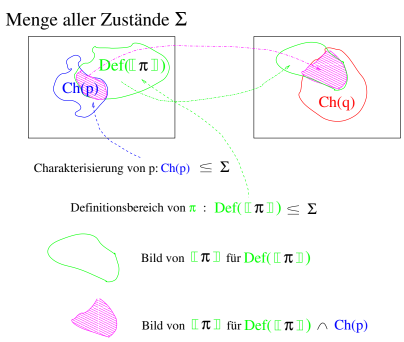
Veranschaulichung (1)

...der Charakterisierung $Ch(p)$ einer logischen Formel p:



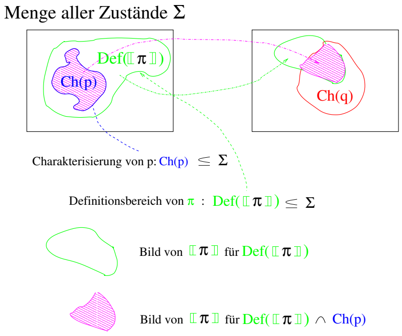
Veranschaulichung (2)

...der Gültigkeit eine Hoareschen Zusicherung $\{p\}\ \pi\ \{q\}$ im Sinne partieller Korrektheit:



Veranschaulichung (3)

...der Gültigkeit eine Hoareschen Zusicherung $[p]\ \pi\ [q]$ im Sinne totaler Korrektheit:



Hoare-Kalkül HK_{PK} für partielle Korrektheit

$$
\begin{array}{rl}
\text{[skip]} & \dfrac{}{\{p\}\ skip\ \{p\}} \\[2ex]
\text{[abort]} & \dfrac{}{\{p\}\ abort\ \{q\}} \\[2ex]
\text{[ass]} & \dfrac{}{\{p[t\backslash x]\}\ x{:=}t\ \{p\}} \\[2ex]
\text{[comp]} & \dfrac{\{p\}\ \pi_1\ \{r\},\ \{r\}\ \pi_2\ \{q\}}{\{p\}\ \pi_1;\pi_2\ \{q\}} \\[2ex]
\text{[ite]} & \dfrac{\{p \wedge b\}\ \pi_1\ \{q\},\ \{p \wedge \neg b\}\ \pi_2\ \{q\}}{\{p\}\ \text{if } b \text{ then } \pi_1 \text{ else } \pi_2 \text{ fi } \{q\}} \\[2ex]
\text{[while]} & \dfrac{\{I \wedge b\}\ \pi\ \{I\}}{\{I\}\ \text{while } b \text{ do } \pi \text{ od } \{I \wedge \neg b\}} \\[2ex]
\text{[cons]} & \dfrac{p \Rightarrow p_1,\ \{p_1\}\ \pi\ \{q_1\},\ q_1 \Rightarrow q}{\{p\}\ \pi\ \{q\}}
\end{array}
$$

Hoare-Kalkül HK_{TK} für totale Korrektheit

...identisch mit HK_{PK}, wobei aber Regel [while] ersetzt ist durch:

$$
\text{[while}_{TK}\text{]} \quad \frac{I \wedge b \Rightarrow u[t/v],\ \{I \wedge b \wedge t{=}w\}\ \pi\ \{I \wedge t{<}w\}}{\{I\}\ \text{while } b \text{ do } \pi \text{ od } \{I \wedge \neg b\}}
$$

wobei

- u Boolescher Ausdruck über der Variablen v,
- t Term,
- w Variable, die in I, b, π und t nicht frei vorkommt,
- $M =_{df} \{\sigma(v) \mid \sigma \in \Sigma\ \wedge\ [\![u]\!]_B(\sigma) = \mathtt{tt}\}$ noethersch geordnete Menge (sog. noethersche Halbordnung).

Korrektheit und Vollständigkeit von HK_{PK} und HK_{TK}

Sei K ein Kalkül für partielle bzw. totale Korrektheit

Zentral sind dann die Fragen der...

- *Korrektheit:* ...ist jede mithilfe von K ableitbare Korrektheitsformel partiell bzw. total korrekt?
- *Vollständigkeit:* ...ist jede partiell bzw. total korrekte Korrektheitsformel mithilfe von K ableitbar?

Speziell:

- Sind HK_{PK} und HK_{TK} *korrekt* und *vollständig*?

Hauptresultate

Zur Korrektheit:

Theorem [Korrektheit von HK_{PK} und HK_{TK}]

1. HK_{PK} ist korrekt, d.h. jede mit HK_{PK} ableitbare Korrektheitsformel ist gültig im Sinne partieller Korrektheit.
2. HK_{TK} ist korrekt, d.h. jede mit HK_{TK} ableitbare Korrektheitsformel ist gültig im Sinne totaler Korrektheit.

Beweis ...durch Induktion über die Anzahl der Regelanwendungen im Beweisbaum zur Ableitung der Korrektheitsformel.

Zur Vollständigkeit:

Für Korrektheitskalküle ist i.a. nur sog. *relative* Vollständigkeit beweisbar. Das gilt auch für HK_{PK} und HK_{TK}. Details dazu später.

Beispiele 1(2)

...Beweis partieller Korrektheit von Hoareschen Zusicherungen anhand zweier Programme zur Berechnung

- der Fakultät und
- der ganzzahligen Division mit Rest

Beispiele 2(2)

Im Detail:

Beweise, dass die beiden Hoareschen Zusicherungen

$$
\begin{array}{c}
\{a > 0\} \\
x := a;\ \ y := 1;\ \ \text{while } x > 1 \text{ do } y := y * x; x := x - 1 \text{ od} \\
\{y = a!\}
\end{array}
$$

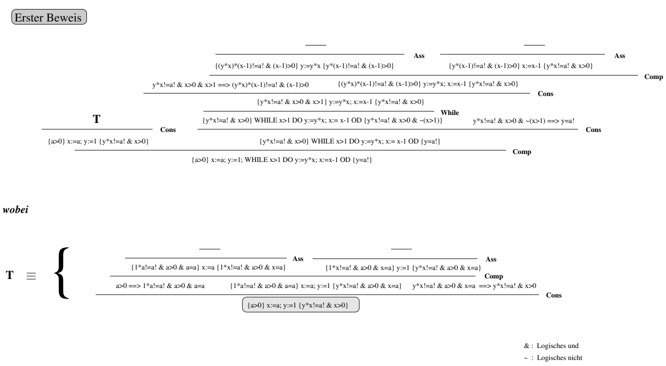
und

$$
\begin{array}{c}
\{x \geq 0 \wedge y > 0\} \\
q := 0; r := x; \text{while } r \geq y \text{ do } q := q + 1; r := r - y \text{ od} \\
\{x = q * y + r \wedge 0 \leq r < y\}
\end{array}
$$

gültig sind im Sinne partieller Korrektheit.

In der Folge geben wir die Beweise dafür in baumartiger Notation an...

Bew. part. Korrektheit: Fakultät (1)



Bew. part. Korrektheit: Fakultät (2)

Zweiter Beweis

```
                                                   Ass                                                   Ass
{(y*x)*(x-1)!=a! & (x-1)>0} y:=y*x {y*(x-1)!=a! & (x-1)>0}     {y*(x-1)!=a! & (x-1)>0} x:=x-1 {y*x!=a! & x>0}
                                                                                                          Comp
y*x!=a! & x>0 & x>1 ==> (y*x)*(x-1)!=a! & (x-1)>0     {(y*x)*(x-1)!=a! & (x-1)>0} y:=y*x; x:=x-1 {y*x!=a! & x>0}
                                                                                                          Cons
{y*x!=a! & x>0 & x>1} y:=y*x; x:=x-1 {y*x!=a! & x>0}
                                                                                                          While
T|                                   {y*x!=a! & x>0} WHILE x>1 DO y:=y*x; x:= x-1 OD {y*x!=a! & x>0 & ~(x>1)}     y*x!=a! & x>0 & ~(x>1) ==> y=a!
                              Cons                                                                        Cons
{a>0} x:=a; y:=1 {y*x!=a! & x>0}     {y*x!=a! & x>0} WHILE x>1 DO y:=y*x; x:= x-1 OD {y=a!}
                                                                                                          Comp
{a>0} x:=a; y:=1; WHILE x>1 DO y:=y*x; x:=x-1 OD {y=a!}
```

wobei

```
                                     Ass                                          Ass
         {1*a!=a! & a>0} x:=a {1*x!=a! & x>0}     {1*x!=a! & x>0} y:=1 {y*x!=a! & x>0}
                                                                                  Comp
T| ≡  {  a>0 ==> 1*a!=a! & a>0     {1*a!=a! & a>0} x:=a; y:=1 {y*x!=a! & x>0}
                                                                                  Cons
         {a>0} x:=a; y:=1 {y*x!=a! & x>0}
```

& : Logisches und
~ : Logisches nicht

Bew. partieller Korrektheit: Division

```
Ass                                                                                                        Ass
{x=(q+1)*y+r-y & r-y>=0} q:=q+1 {x=q*y+r-y & r-y>=0}     {x=q*y+r-y & r-y>=0} r:=r-y {x=q*y+r & r>=0}
Comp
(x=q*y+r & r>=0 & r>=y) => (x=(q+1)*y+r-y & r-y>=0)   {x=(q+1)*y+r-y & r-y>=0} q:=q+1; r:=r-y {x=q*y+r & r>=0}
Cons
{x=q*y+r & r>=0 & r>=y} q:=q+1; r:=r-y {x=q*y+r & r>=0}
While
{x=q*y+r & r>=0} WHILE r>=y DO q:=q+1; r:=r-y OD {x=q*y+r & r>=0 & ~(r>=y)}   (x=q*y+r & r>=0 & ~(r>=y)) => (x=q*y+r & r>=0 & r<y)
Cons

Ass                                                                           Ass
{x=0*y+x & x>=0} q:=0 {x=q*y+x & x>=0}   {x=q*y+x & x>=0} r:=x {x=q*y+r & r>=0}
                                                                              Comp
(x>=0 & y>0) => (x=0*y+x & x>=0) {x=0*y+x & x>=0} q:=0; r:=x {x=q*y+r & r>=0}
                                                                              Cons
{x>=0 & y>0} q:=0; r:=x {x=q*y+r & r>=0}          {x=q*y+r & r>=0} WHILE r>=y DO q:=q+1; r:=r-y OD {x=q*y+r & r>=0 & r<y}
                                                                                                                    Comp
{x>=0 & y>0} q:=0; r:=x; WHILE r>=y DO q:=q+1; r:=r-y OD {x=q*y+r & r>=0 & r<y}
```

& : Logisches und
~ : Logisches nicht

Bew. part. Korrektheit: Fakultät (1)

- Die unmittelbare baumartige Notation von Hoareschen Korrektheitsbeweisen ist i.a. unhandlich.
- Alternativ hat sich deshalb eine Notationsvariante eingebürgert, bei der in den Programmtext Zusicherungen als Annotationen eingestreut werden.
- In der Folge demonstrieren wir diesen Notationsstil am Beispiel des Nachweises der partiellen Korrektheit unseres Fakultätsprogramms bezüglich der angegebenen Vor- und Nachbedingung. Man spricht auch von einem sog. *linearen Beweis* bzw. *linearen Beweisskizze*.

Bew. part. Korrektheit: Fakultät (2)

Beweise, dass das Hoare-Tripel

$$\{a > 0\}$$
$$x := a;\ y := 1;\ \text{while } x > 1 \text{ do } y := y * x; x := x - 1 \text{ od}$$
$$\{y = a!\}$$

gültig ist im Sinne partieller Korrektheit.

Wir entwickeln den Beweis in der Folge Schritt für Schritt!

Bew. part. Korrektheit: Fakultät (3)

Schritt 1

"Träumen" der Invariante...

- $\{y * x! = a! \wedge x > 0\}$

...um die [while]-Regel anwenden zu können.

Bew. part. Korrektheit: Fakultät (4)

Schritt 2

Behandlung des Rumpfs der while-Schleife...

Der Nachweis der Gültigkeit von

$$\begin{array}{c} \{y * x! = a! \wedge x > 0 \wedge x > 1\} \\ y := y * x; \\ x := x - 1; \\ \{y * x! = a! \wedge x > 0\} \end{array}$$

erlaubte mithilfe der [while]-Regel den Übergang zu:

$$\begin{array}{c} \{y * x! = a! \wedge x > 0\} \\ \text{while } x > 1 \text{ do} \\ \{y * x! = a! \wedge x > 0 \wedge x > 1\} \\ y := y * x; \\ x := x - 1; \\ \{y * x! = a! \wedge x > 0\} \\ \text{od [while]} \\ \{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\} \end{array}$$

Bew. part. Korrektheit: Fakultät (5)

Behandlung des Rumpfs der while-Schleife im Detail:

$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$

$$\begin{array}{c} y := y * x; \\ x := x - 1; \\ \{y * x! = a! \wedge x > 0\} \end{array}$$

Bew. part. Korrektheit: Fakultät (6)

Wegen *Rückwärts*zuweisungsregel wird der Rumpf der while-Schleife von hinten nach vorne bearbeitet:

$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$

$$\begin{array}{c} y := y * x; \\ \{y * (x-1)! = a! \wedge x - 1 > 0\} \\ x := x - 1;\ \text{[ass]} \\ \{y * x! = a! \wedge x > 0\} \end{array}$$

Bew. part. Korrektheit: Fakultät (7)

Nach abermaliger Anwendung der [ass]-Regel erhalten wir...

$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$

$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$

...wobei noch eine “Beweislücke” verbleibt!

Bew. part. Korrektheit: Fakultät (8)

Schluss der “Beweislücke” in der zugrundeliegenden Theorie:

$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$

Bew. part. Korrektheit: Fakultät (9)

Anwendung der [while]-Regel liefert nun wie gewünscht:

$$\{y * x! = a! \wedge x > 0\}$$
$$\text{while } x > 1 \text{ do}$$
$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{od [while]}$$
$$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$$

Bew. part. Korrektheit: Fakultät (10)

Schritt 3

Zur gewünschten Nachbedingung verbleibt offenbar ebenfalls eine Beweislücke:

$$\{y * x! = a! \wedge x > 0\}$$
$$\text{while } x > 1 \text{ do}$$
$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{od [while]}$$
$$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$$

$$\{y = a!\}$$

Bew. part. Korrektheit: Fakultät (11)

Schluss der Beweislücke in der zugrundeliegenden Theorie:

$$\{y * x! = a! \wedge x > 0\}$$
$$\text{while } x > 1 \text{ do}$$
$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{od [while]}$$
$$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$$
$$\Downarrow \text{ [cons]}$$
$$\{y * x! = a! \wedge x > 0 \wedge x \leq 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{y * x! = a! \wedge x = 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{y = a!\}$$

Bew. part. Korrektheit: Fakultät (12)

Aus Platzgründen etwas verkürzt dargestellt:

$$\{y * x! = a! \wedge x > 0\}$$
$$\text{while } x > 1 \text{ do}$$
$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{od [while]}$$
$$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$$
$$\Downarrow \text{ [cons]}$$
$$\{y = a!\}$$

Bew. part. Korrektheit: Fakultät (13)

Schritt 4

Es verbleibt, die Beweislücke zur gewünschten Vorbedingung zu schließen:

$$\{a > 0\}$$

$$x := a;$$
$$y := 1;$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{while } x > 1 \text{ do}$$
$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{od [while]}$$
$$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$$
$$\Downarrow \text{ [cons]}$$
$$\{y = a!\}$$

Bew. part. Korrektheit: Fakultät (14)

Einmalige Anwendung der [ass]-Regel liefert:

$$\{a > 0\}$$

$$x := a;$$
$$\{1 * x! = a! \wedge x > 0\}$$
$$y := 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{while } x > 1 \text{ do}$$
$$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$$
$$\Downarrow \text{ [cons]}$$
$$\{(y * x) * (x-1)! = a! \wedge x - 1 > 0\}$$
$$y := y * x; \text{ [ass]}$$
$$\{y * (x-1)! = a! \wedge x - 1 > 0\}$$
$$x := x - 1; \text{ [ass]}$$
$$\{y * x! = a! \wedge x > 0\}$$
$$\text{od [while]}$$
$$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$$
$$\Downarrow \text{ [cons]}$$
$$\{y = a!\}$$

Bew. part. Korrektheit: Fakultät (15)

Abermalige Anwendung der [ass]-Regel liefert:

$\{a > 0\}$

$\{1 * a! = a! \wedge a > 0\}$
$x := a;$ [ass]
$\{1 * x! = a! \wedge x > 0\}$
$y := 1;$ [ass]
$\{y * x! = a! \wedge x > 0\}$
while $x > 1$ do
$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$
$\Downarrow$ [cons]
$\{(y * x) * (x - 1)! = a! \wedge x - 1 > 0\}$
$y := y * x;$ [ass]
$\{y * (x - 1)! = a! \wedge x - 1 > 0\}$
$x := x - 1;$ [ass]
$\{y * x! = a! \wedge x > 0\}$
od [while]
$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$
$\Downarrow$ [cons]
$\{y = a!\}$

Bew. part. Korrektheit: Fakultät (16)

Schluss der letzten Beweislücke in der zugrundeliegenden Theorie:

$\{a > 0\}$
$\Downarrow$ [cons]
$\{1 * a! = a! \wedge a > 0\}$
$x := a;$ [ass]
$\{1 * x! = a! \wedge x > 0\}$
$y := 1;$ [ass]
$\{y * x! = a! \wedge x > 0\}$
while $x > 1$ do
$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$
$\Downarrow$ [cons]
$\{(y * x) * (x - 1)! = a! \wedge x - 1 > 0\}$
$y := y * x;$ [ass]
$\{y * (x - 1)! = a! \wedge x - 1 > 0\}$
$x := x - 1;$ [ass]
$\{y * x! = a! \wedge x > 0\}$
od [while]
$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$
$\Downarrow$ [cons]
$\{y = a!\}$

Überblick (17)

$\{a > 0\}$
$\Downarrow$ [cons]
$\{1 * a! = a! \wedge a > 0\}$
$x := a;$ [ass]
$\{1 * x! = a! \wedge x > 0\}$
$y := 1;$ [ass]
$\{y * x! = a! \wedge x > 0\}$
while $x > 1$ do
$\{y * x! = a! \wedge x > 0 \wedge x > 1\}$
$\Downarrow$ [cons]
$\{(y * x) * (x - 1)! = a! \wedge x - 1 > 0\}$
$y := y * x;$ [ass]
$\{y * (x - 1)! = a! \wedge x - 1 > 0\}$
$x := x - 1;$ [ass]
$\{y * x! = a! \wedge x > 0\}$
od [while]
$\{y * x! = a! \wedge x > 0 \wedge \neg(x > 1)\}$
$\Downarrow$ [cons]
$\{y * x! = a! \wedge x > 0 \wedge x \leq 1\}$
$\Downarrow$ [cons]
$\{y * x! = a! \wedge x = 1\}$
$\Downarrow$ [cons]
$\{y = a!\}$

Bew. part. Korrektheit: Fakultät (18)

Damit haben wir insgesamt wie gewünscht gezeigt:

Das Hoaresche Tripel

$$\{a > 0\}$$
$$x := a;\ y := 1;\ \text{while } x > 1 \text{ do } y := y * x; x := x - 1 \text{ od}$$
$$\{y = a!\}$$

ist gültig im Sinne partieller Korrektheit.

Linearer vs. baumartiger Beweisstil

Vorteil linearen gegenüber baumartigen Beweisnotationsstils:

- wenig Redundanz
- daher insgesamt knappere Beweise

Sprechweisen im Zshg. mit Hoare-Tripeln (1)

Hoaresche Zusicherungen sind von einer der zwei Formen

- $\{p\}\ \pi\ \{q\}$ und
- $[p]\ \pi\ [q]$

wobei

- p, q logische Formeln sind (meist prädikatenlogische Formeln 1. Stufe) und
- π ein Programm ist.

Sprechweisen im Zshg. mit Hoare-Tripeln (2)

In einer Hoareschen Zusicherung von einer der Formen

- $\{p\}\ \pi\ \{q\}$ und
- $[p]\ \pi\ [q]$

heißen

- p und q *Vor-* bzw. *Nachbedingung*.

Sprechweisen im Zshg. mit Hoare-Tripeln (3)

In einer Hoareschen Zusicherung werden üblicherweise

- geschweifte Klammern wie in
  $\{p\}\ \pi\ \{q\}$
  für Tripel im Sinne *partieller Korrektheit* und
- eckige Klammern wie in
  $[p]\ \pi\ [q]$
  für Tripel im Sinne *totaler Korrektheit*

benutzt.

Sprechweisen im Zshg. mit Hoare-Tripeln (4)

Zwei Beispiele Hoarescher Zusicherungen:

$$\{a > 0\}$$
$$x := a;\ y := 1;\ \text{while } x > 1 \text{ do } y := y * x; x := x - 1 \text{ od}$$
$$\{y = a!\}$$

...zum Ausdruck partieller Korrektheit von π bzgl. der Vorbedingung $a > 0$ und der Nachbedingung $y = a!$

$$[a > 0]$$
$$x := a;\ y := 1;\ \text{while } x > 1 \text{ do } y := y * x; x := x - 1 \text{ od}$$
$$[y = a!]$$

...zum Ausdruck totaler Korrektheit von π bzgl. der Vorbedingung $a > 0$ und der Nachbedingung $y = a!$

Sprechweisen im Zshg. mit Hoare-Tripeln (5)

Die Wortwahl

- *Hoaresches Tripel* oder kurz *Hoare-Tripel* bzw.
- *Hoaresche Zusicherung* oder kurz *Korrektsheitsformel*

betont jeweils die

- syntaktische bzw.
- semantische Sicht

auf

- $\{p\}\ \pi\ \{q\}$ bzw. $[p]\ \pi\ [q]$