Theoretische Informatik
Übungsblatt 5 (2024W)
Lösungsvorschlag

In den ersten sechs Aufgaben steht Σ für das folgende Programm.

$x \coloneqq x - y$;
if $x > 0$ **then**
  $y \coloneqq x + y$
else
  abort
fi

Aufgabe 5.1
Zeigen Sie, dass das Programm Σ nach Hinzufügen einiger Klammernpaare ein syntaktisch korrektes SIMPLE($\mathbb{Z}$)-Programm ist.
Hinweis: Geben Sie eine Parallelableitung an.

Lösung 5.1
Bei einer Parallelableitung werden ausgehend von der Startvariablen in jedem Ableitungsschritt alle Nonterminale mittels einer geeigneten Produktion ersetzt.

Programm
$\Rightarrow_P$ *Programm* „**;** “ *Programm*
$\Rightarrow_P$ *Var* „:=“ *Term* „**; if**“ *Term* „**then**“ *Programm* „**else**“ *Programm* „**fi**“
$\Rightarrow_P$ „$x \coloneqq$ (“ *Term BinOp Term* „); **if** (“ *Term BinOp Term* „) **then**“ *Var* „:=“ *Term* „**else abort fi**“
$\Rightarrow_P$ „$x \coloneqq$ (“ *Var* „$-$“ *Var* „); **if** (“ *Var* „$>$“ *Const* „) **then** $y \coloneqq$ (“ *Term BinOp Term* „) **else abort fi**“
$\Rightarrow_P$ „$x \coloneqq (x - y)$; **if** $(x > 0)$ **then** $y \coloneqq$ (“ *Var* „$+$“ *Var* „) **else abort fi**“
$\Rightarrow_P$ „$x \coloneqq (x - y)$; **if** $(x > 0)$ **then** $y \coloneqq (x + y)$ **else abort fi**“

Aufgabe 5.2
Sei σ ein Zustand mit $\sigma(x) = 1$ und $\sigma(y) = 0$. Berechnen Sie $[\Sigma]\sigma$ mittels der strukturellen operationalen Semantik von SIMPLE($\mathbb{Z}$).

Lösung 5.2
Um das Ergebnis des Programms, $[\Sigma]\sigma$, zu erhalten, müssen wir mit der Konfiguration (Σ, σ) beginnen und solange Rechenschritte durchführen, bis wir eine finale Konfiguration erreichen.

$$
\begin{aligned}
(\Sigma, \sigma) &= (x \coloneqq x{-}y \text{; } \textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi}, \sigma) \\
&\quad (x \coloneqq x{-}y, \sigma) \Rightarrow \sigma_1, \text{ wobei } \sigma_1(x) = [x{-}y]\sigma = [x]\sigma - [y]\sigma = \sigma(x) - \sigma(y) = 1 - 0 = 1 \\
&\qquad\qquad \sigma_1(v) = \sigma(v) \quad \text{für } v \neq x \\
&\Rightarrow (\textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi}, \sigma_1) \\
&\quad [x > 0]\sigma_1 = ([x]\sigma_1 > [0]\sigma_1) = (\sigma_1(x) > 0) = (1 > 0) = 1 \\
&\Rightarrow (y \coloneqq x{+}y, \sigma_1) \\
&\Rightarrow \sigma_2, \text{ wobei } \sigma_2(y) = [x{+}y]\sigma_1 = [x]\sigma_1 + [y]\sigma_1 = \sigma_1(x) + \sigma_1(y) = 1 + 0 = 1 \\
&\qquad\qquad \sigma_2(v) = \sigma_1(v) \quad \text{für } v \neq y
\end{aligned}
$$

Somit gilt $[\Sigma]\sigma = \sigma_2$, da $(\Sigma, \sigma) \overset{*}{\Rightarrow} \sigma_2$.

Aufgabe 5.3
Sei σ ein Zustand mit $\sigma(x) = 0$ und $\sigma(y) = 0$. Berechnen Sie $[\Sigma]\sigma$ mittels der natürlichen Semantik von SIMPLE($\mathbb{Z}$).

Lösung 5.3

Die natürliche Semantik erlaubt es, $[\Sigma]\,\sigma$ direkt zu berechnen.

$$\begin{aligned}
[\Sigma]\,\sigma &= [x \coloneqq x{-}y;\ \textbf{if}\ x{>}0\ \textbf{then}\ y \coloneqq x{+}y\ \textbf{else abort fi}]\,\sigma \\
&= [\textbf{if}\ x{>}0\ \textbf{then}\ y \coloneqq x{+}y\ \textbf{else abort fi}]\,[x \coloneqq x{-}y]\,\sigma \\
&= [\textbf{if}\ x{>}0\ \textbf{then}\ y \coloneqq x{+}y\ \textbf{else abort fi}]\,\sigma_1, \text{ wobei } \sigma_1(x) = [x{-}y]\sigma = [x]\sigma - [y]\sigma = 0 - 0 = 0 \\
&\qquad\qquad \sigma_1(v) = \sigma(v) \quad \text{für } v \neq x \\
&\qquad [x > 0]\sigma_1 = ([x]\sigma_1 > [0]\sigma_1) = (0 > 0) = 0 \\
&= [\textbf{abort}]\,\sigma_1
\end{aligned}$$

$[\textbf{abort}]\,\sigma_1$ ist undefiniert, daher ist es auch $[\Sigma]\sigma$.

Aufgabe 5.4

Zeigen Sie, dass die Aussage $\{x \geq y \wedge x = x_0\}\,\Sigma\,\{y = x_0\}$ partiell korrekt ist, indem Sie die Korrektheitsaussage mit den Regeln und Axiomen des Hoare-Kalküls ableiten. Argumentieren Sie, warum die auftretenden Formeln gültig sind.

Zeigen Sie weiteres, dass die Korrektheitsaussage nicht total korrekt ist, indem Sie ein Gegenbeispiel angeben.

Lösung 5.4

Wir zerlegen die Korrektheitsaussage nach den Regeln des Hoare-Kalküls. Die Ableitung lässt sich von oben nach unten als eine Reihe von Regelanwendungen lesen, bei denen man von wahren Korrektheitsaussagen und gültigen Formeln auf neue wahre Korrektheitsaussagen schließt. Konstruiert wird die Ableitung aber von unten nach oben, beginnend mit der zu beweisenden Korrektheitsaussage.

$$\dfrac{\dfrac{\{x \geq y \wedge x = x_0\}\,x \coloneqq x{-}y\,\{G\}\ \text{(zw')} \qquad \dfrac{\dfrac{G \wedge x{>}0 \Rightarrow (y = x_0)\left[\begin{smallmatrix}x+y\\ y\end{smallmatrix}\right] \quad \{(y = x_0)\left[\begin{smallmatrix}x+y\\ y\end{smallmatrix}\right]\}\,y \coloneqq x{+}y\,\{y = x_0\}\ \text{(zw)}}{\{G \wedge x{>}0\}\,y \coloneqq x{+}y\,\{y = x_0\}}\ \text{(imp)} \quad \{G \wedge x{\not>}0\}\ \textbf{abort}\ \{y = x_0\}\ \text{(ab)}}{\{G\}\ \textbf{if}\ x{>}0\ \textbf{then}\ y \coloneqq x{+}y\ \textbf{else abort fi}\ \{y = x_0\}}\ \text{(if)}}{\{x \geq y \wedge x = x_0\}\,x \coloneqq x{-}y;\ \textbf{if}\ x{>}0\ \textbf{then}\ y \coloneqq x{+}y\ \textbf{else abort fi}\ \{y = x_0\}}\ \text{(ha)}$$

Wir wählen G so, dass $\{x \geq y \wedge x = x_0\}\,x \coloneqq x{-}y\,\{G\}$ zu einer Instanz des Axioms zw′ wird, d.h., wir wählen

$$G = \exists x'(x' \geq y \wedge x' = x_0 \wedge x = x'{-}y)\ .$$

Somit ist die Gültigkeit der folgenden Implikation zu zeigen.

$$\begin{aligned}
&G \wedge x{>}0 \Rightarrow (y = x_0)\left[\begin{smallmatrix}x+y\\ y\end{smallmatrix}\right] \\
&\exists x'(x' \geq y \wedge x' = x_0 \wedge x = x'{-}y) \wedge x{>}0 \Rightarrow x{+}y = x_0
\end{aligned}$$

Wir formen das letzte Konjunkt im Bereich des Existenzquantors zu $x{+}y = x'$ um und können damit die übrigen Vorkommnisse von x' durch $x{+}y$ ersetzen.

$$\exists x'(x{+}y \geq y \wedge x{+}y = x_0 \wedge x{+}y = x') \wedge x{>}0 \Rightarrow x{+}y = x_0$$

Wir können nun $x{+}y = x_0$ vor den Quantor ziehen, da die gebundene Variable x' in der Formel nicht vorkommt.

$$x{+}y = x_0 \wedge \exists x'(x{+}y \geq y \wedge x{+}y = x') \wedge x{>}0 \Rightarrow x{+}y = x_0$$

Diese Formel ist eine Tautologie der Form $F \wedge \cdots \Rightarrow F$, also gültig. Somit ist die Korrektheitsaussage in der letzten Zeile der Ableitung mit den Regeln des Hoare-Kalküls für partielle Korrektheit aus den Axiomen zw′, zw, ab und einer gültigen Formel mit den Regeln des Hoare-Kalküls für partielle Korrektheit ableitbar, sie ist daher partiell korrekt.

Um zu zeigen, dass die Korrektheitsaussage nicht total korrekt ist, benötigen wir eine zulässige Eingabe, für die das Programm entweder kein Ergebnis liefert oder das Ergebnis nicht die Nachbedingung erfüllt. Da die Aussage partiell korrekt ist, erfüllt jedes Ergebnis einer zulässigen Eingabe die Nachbedingung. Wir benötigen daher eine zulässige Eingabe σ, für die das Ergebnis des Programms nicht definiert ist. Wir wählen $\sigma(x) = \sigma(y) = \sigma(x_0) = 0$ und erhalten:

- σ ist eine zulässige Eingabe, da sie die Vorbedingung erfüllt.

$$[x{\geq}y \wedge x{=}x_0]\,\sigma = (\sigma(x){\geq}\sigma(y) \wedge \sigma(x){=}\sigma(x_0)) = (0{\geq}0 \wedge 0{=}0) = 1$$

- Das Ergebnis $[\Sigma]\,\sigma$ ist nicht definiert.

$$[\Sigma]\,\sigma = \cdots = [\textbf{abort}]\,\sigma_1 = \text{undefiniert, siehe Aufgabe 3}$$

Die Korrektheitsaussage ist daher nicht wahr bzgl. totaler Korrektheit, ist also nicht „total korrekt“.

Aufgabe 5.5

Zeigen Sie, dass die Aussage $\{x \geq y \wedge x = x_0\}\,\Sigma\,\{y = x_0\}$ partiell korrekt ist, indem Sie wahlweise die schwächste liberale Vorbedingung (wlp) oder die stärkste Nachbedingung (sp) des Programms berechnen. Argumentieren Sie, warum die auftretende Implikation gültig ist.

Lösung 5.5

Schwächste liberale Vorbedingungen: Wir berechnen zuerst $\mathsf{wlp}(\Sigma, y{=}x_0)$ und zeigen danach, dass diese Formel von der Vorbedingung impliziert wird.

$$\begin{aligned}
\mathsf{wlp}(\Sigma, y{=}x_0) &= \mathsf{wlp}(x \coloneqq x{-}y\textbf{; if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi}, y{=}x_0)\\
&= \mathsf{wlp}(x \coloneqq x{-}y,\ \mathsf{wlp}(\textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi}, y{=}x_0))\\
&= \mathsf{wlp}(x \coloneqq x{-}y,\ (x{>}0 \Rightarrow \mathsf{wlp}(y \coloneqq x{+}y, y{=}x_0)) \wedge (x{\not>}0 \Rightarrow \mathsf{wlp}(\textbf{abort}, y{=}x_0)))\\
&= \mathsf{wlp}(x \coloneqq x{-}y,\ (x{>}0 \Rightarrow x{+}y{=}x_0) \wedge (x{\not>}\,0 \Rightarrow 1))\\
&= \mathsf{wlp}(x \coloneqq x{-}y,\ (x{>}0 \Rightarrow x{+}y{=}x_0) \wedge 1)\\
&= \mathsf{wlp}(x \coloneqq x{-}y,\ (x{>}0 \Rightarrow x{+}y{=}x_0))\\
&= (x{-}y{>}0 \Rightarrow x{-}y{+}y{=}x_0)\\
&= (x{>}y \Rightarrow x{=}x_0)\\
&= (x{\not>}y \vee x{=}x_0)
\end{aligned}$$

$$\begin{aligned}
&(x{\geq}y \wedge x{=}x_0) \Rightarrow \mathsf{wlp}(\Sigma, y{=}x_0)\\
&(x{\geq}y \wedge x{=}x_0) \Rightarrow (x{\not>}y \vee x{=}x_0)
\end{aligned}$$

Die Implikation in der letzten Zeile ist eine Tautologie der Form $\cdots \wedge F \Rightarrow \cdots \vee F$ und daher gültig.

Stärkste Nachbedingung: Wir berechnen zuerst $\mathsf{sp}(x{\geq}y{\wedge}x{=}x_0, \Sigma)$ und zeigen dann, dass diese Formel die Nachbedingung impliziert.

$$\begin{aligned}
\mathsf{sp}(x{\geq}y \wedge x{=}x_0, \Sigma) &= \mathsf{sp}(x{\geq}y \wedge x{=}x_0,\ x \coloneqq x{-}y\textbf{; if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi})\\
&= \mathsf{sp}(\mathsf{sp}(x{\geq}y \wedge x{=}x_0,\ x \coloneqq x{-}y),\ \textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi})\\
&= \mathsf{sp}(\exists x'(x'{\geq}y \wedge x'{=}x_0 \wedge x{=}x'{-}y),\ \textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi})\\
&= \mathsf{sp}(\exists x'(x{+}y{\geq}y \wedge x{+}y{=}x_0 \wedge x'{=}x{+}y),\ \textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi})\\
&= \mathsf{sp}(x{+}y{\geq}y \wedge x{+}y{=}x_0 \wedge \exists x' x'{=}x{+}y,\ \textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi})\\
&= \mathsf{sp}(x{\geq}0 \wedge x{+}y{=}x_0 \wedge 1,\ \textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi})\\
&= \mathsf{sp}(x{\geq}0 \wedge x{+}y{=}x_0,\ \textbf{if } x{>}0 \textbf{ then } y \coloneqq x{+}y \textbf{ else abort fi})\\
&= \mathsf{sp}(x{>}0 \wedge x{\geq}0 \wedge x{+}y{=}x_0, y \coloneqq x{+}y) \vee \mathsf{sp}(x{\not>}0 \wedge x{\geq}0 \wedge x{+}y{=}x_0, \textbf{abort})\\
&= \mathsf{sp}(x{>}0 \wedge x{+}y{=}x_0, y \coloneqq x{+}y) \vee 0\\
&= \mathsf{sp}(x{>}0 \wedge x{+}y{=}x_0, y \coloneqq x{+}y)\\
&= \exists y'(x{>}0 \wedge x{+}y'{=}x_0 \wedge y{=}x{+}y')\\
&= \exists y'(x{>}0 \wedge x{+}y{-}x{=}x_0 \wedge y'{=}y{-}x)\\
&= x{>}0 \wedge y{=}x_0 \wedge \exists y' y'{=}y{-}x\\
&= x{>}0 \wedge y{=}x_0 \wedge 1\\
&= x{>}0 \wedge y{=}x_0
\end{aligned}$$

$$\begin{aligned}
&\mathsf{sp}(x{\geq}y \wedge x{=}x_0, \Sigma) \Rightarrow y{=}x_0\\
&\qquad x{>}0 \wedge y{=}x_0 \Rightarrow y{=}x_0
\end{aligned}$$

Die Implikation in der letzten Zeile ist eine Tautologie der Form $\cdots \wedge F \Rightarrow F$ und daher gültig.

Aufgabe 5.6

Zeigen Sie, dass die Aussage $\{x > y \wedge x = x_0\}\,\Sigma\,\{y = x_0\}$ total korrekt ist. Führen Sie den Korrektheitsbeweis, indem Sie das Programm mithilfe der Annotierungsregeln um Zustandsbeschreibungen ergänzen und die Gültigkeit der auftretenden Implikationen argumentieren.

Lösung 5.6

Wir beginnen mit der zu beweisenden Korrektheitsaussage und fügen schrittweise Zustandsbeschreibungen hinzu. Wir nummerieren die Formeln in der Reihenfolge des Hinzufügens.

$\{\, \mathit{Pre}\colon x > y \wedge x = x_0 \,\}$
zw↑ $\{\, F_7\colon x{-}y > 0 \wedge x{-}y{+}y = x_0 \,\}$
$x \coloneqq x - y$**;**
$\{\, F_6'\colon x{>}0 \wedge x{+}y = x_0 \,\}$
if↑ $\{\, F_6\colon (x{>}0 \wedge x{+}y = x_0) \vee (x{\not>}0 \wedge 0) \,\}$
if $x > 0$ **then**
zw↑ $\quad\{\, F_5\colon y = x_0 \left[\begin{smallmatrix} x+y \\ y \end{smallmatrix}\right] \,\}$
$\quad y \coloneqq x + y$
fi↑ $\quad\{\, F_3\colon y = x_0 \,\}$
else
ab $\quad\{\, F_1\colon 0 \,\}$
$\quad$**abort**
ab $\quad\{\, F_2\colon 0 \,\}$
fi↑ $\quad\{\, F_4\colon y = x_0 \,\}$
fi
$\{\, \mathit{Post}\colon y = x_0 \,\}$

Die Formel F_6' ist lediglich eine Vereinfachung der Formel F_6 und logisch äquivalent zu dieser.
Es ist die Gültigkeit von zwei Implikationen zu zeigen.

- $\mathit{Pre} \Rightarrow F_7$:
  $x > y \wedge x = x_0 \Rightarrow x{-}y > 0 \wedge x{-}y{+}y = x_0$
  $x > y \wedge x = x_0 \Rightarrow x > y \wedge x = x_0$
  Die letzte Implikation ist eine Tautologie der Form $A \Rightarrow A$ und damit gültig.
- $F_2 \Rightarrow F_4$, d.h., $0 \Rightarrow y{=}x_0$. Aus Falschem folgt Beliebiges, daher ist diese Implikation ebenfalls gültig.

Damit haben wir die totale Korrektheit mit Hilfe des Annotierungskalküls gezeigt.

Aufgabe 5.7

Geben Sie ein Programm Π an, das die drei folgenden Bedingungen erfüllt.

$$\mathsf{wp}(\Pi, 1) = (x \geq y)$$
$$\mathsf{wlp}(\Pi, x = y) = (x < y \vee y = 0)$$
$$\mathsf{sp}(x = y, \Pi) = (x = 0)$$

Verifizieren Sie, dass Ihr Programm diese Bedingungen erfüllt, indem Sie die Ausdrücke links für Ihr Programm berechnen und das Ergebnis mit den angegebenen Formeln vergleichen.

Lösung 5.7

Aus $\mathsf{wp}(\Pi, 1) = (x \geq y)$ können wir schließen, dass das Ergebnis von Π nur für Eingaben mit $x \geq y$ definiert ist. Das trifft z.B. auf die Programme

if $x \geq y$ **then** Ω **else abort fi**
if $x \geq y$ **then skip else abort fi;** Ω

zu, wobei Ω jedes Programm sein kann, das zumindest für $x \geq y$ ein Ergebnis liefert.
Da Π nur für $x \geq y$ terminiert und $\mathsf{wlp}(\Pi, x = y) = (x < y \vee y = 0)$ gelten soll, müssen Eingaben, für deren Ausgabe $x = y$ gilt, beides erfüllen: $x \geq y \wedge (x < y \vee y = 0)$. Diese Bedingung lässt sich zu

$x \geq 0 \wedge y = 0$ vereinfachen. Wir können also für Ω beispielsweise das Programm $x \coloneqq 0$ oder $x \coloneqq y$ wählen.
Die dritte Bedingung, $\mathsf{sp}(x = y, \Pi) = (x = 0)$, legt fest, dass für Eingaben mit $x = y$ das Programm entweder nicht terminiert oder das Ergebnis die Bedingung $x = 0$ erfüllt. Das legt $\Omega = x \coloneqq 0$ nahe. Wir wählen daher für Π das Programm

$$\textbf{if } x \geq y \textbf{ then } x \coloneqq 0 \textbf{ else abort fi}$$

Damit erhalten wir für die drei Bedingungen:

$$\begin{aligned}
\mathsf{wp}(\Pi, 1) &= (x \geq y \wedge \mathsf{wp}(x \coloneqq 0, 1)) \vee (x < y \wedge \mathsf{wp}(\textbf{abort}, 1)) \\
&= (x \geq y \wedge 1) \vee (x < y \wedge 0) \\
&= (x \geq y) \\
\mathsf{wlp}(\Pi, x = y) &= (x \geq y \wedge \mathsf{wlp}(x \coloneqq 0, x = y)) \vee (x < y \wedge \mathsf{wlp}(\textbf{abort}, x = y)) \\
&= (x \geq y \wedge 0 = y) \vee (x < y \wedge 1) \\
&= (x \geq y \wedge 0 = y) \vee x < y \\
&= (x < y \vee y = 0) \\
\mathsf{sp}(x = y, \Pi) &= \mathsf{sp}(x = y \wedge x \geq y, x \coloneqq 0) \vee \mathsf{sp}(x = y \wedge x < y, \textbf{abort}) \\
&= \mathsf{sp}(x = y, x \coloneqq 0) \vee 0 \\
&= \exists x'(x' = y \wedge x = 0) \\
&= (x = 0 \wedge \exists x' x' = y) \\
&= (x = 0 \wedge 1) \\
&= (x = 0)
\end{aligned}$$

Aufgabe 5.8
Zeigen Sie, dass die folgende Aussage total korrekt ist.

$\{\, n \geq 0 \,\}$
$i \coloneqq 0$**;**
$s \coloneqq 0$**;**
while $i \leq n$ **do**
  $s \coloneqq s + 2i$**;**
  $i \coloneqq i + 1$
od;
$\{\, s = n^2 + n \,\}$

Hinweis: Als Invariante ist die Formel $s = i(i-1) \wedge 0 \leq i \leq n + 1$ geeignet, als Variante der Ausdruck $n + 1 - i$.

Lösung 5.8
Wir annotieren die Korrektheitsaussage schrittweise mit Zustandsbeschreibungen innerhalb des Programms. Wir nummerieren die Formeln in der Reihenfolge des Hinzufügens.

	$\{\, \mathit{Pre}\colon n \geq 0 \,\}$
zw↑	$\{\, F_8\colon \mathit{Inv}\left[\begin{smallmatrix}0\\ s\end{smallmatrix}\right]\left[\begin{smallmatrix}0\\ i\end{smallmatrix}\right] \,\}$
	$i \coloneqq 0$**;**
zw↑	$\{\, F_7\colon \mathit{Inv}\left[\begin{smallmatrix}0\\ s\end{smallmatrix}\right] \,\}$
	$s \coloneqq 0$**;**
wht	$\{\, F_1\colon \mathit{Inv} \,\}$
	while $i \leq n$ **do**
wht	$\quad \{\, F_2\colon \mathit{Inv} \wedge i \leq n \wedge t = t_0 \,\}$
zw↑	$\quad \{\, F_6\colon (\mathit{Inv} \wedge 0 \leq t < t_0)\left[\begin{smallmatrix}i+1\\ i\end{smallmatrix}\right]\left[\begin{smallmatrix}s+2i\\ s\end{smallmatrix}\right] \,\}$
	$\quad s \coloneqq s + 2i$**;**
zw↑	$\quad \{\, F_5\colon (\mathit{Inv} \wedge 0 \leq t < t_0)\left[\begin{smallmatrix}i+1\\ i\end{smallmatrix}\right] \,\}$
	$\quad i \coloneqq i + 1$
wht	$\quad \{\, F_3\colon \mathit{Inv} \wedge 0 \leq t < t_0 \,\}$
	od;
wht	$\{\, F_4\colon \mathit{Inv} \wedge i \not\leq n \,\}$
	$\{\, \mathit{Post}\colon s = n^2 + n \,\}$

Nun müssen wir noch die Gültigkeit der Formeln $Pre \Rightarrow F_8$, $F_2 \Rightarrow F_6$ und $F_4 \Rightarrow Post$ argumentieren, wobei wir Invariante und Variante gemäß dem Hinweis in der Angabe wählen.

- $Pre \Rightarrow F_8$:

  $n \geq 0 \Rightarrow Inv\begin{bmatrix}0\\s\end{bmatrix}\begin{bmatrix}0\\i\end{bmatrix}$

  $n \geq 0 \Rightarrow 0 = 0(0-1) \wedge 0 \leq 0 \leq n+1$

  Wir müssen zeigen, dass die Bedingungen auf der rechten Seite erfüllt sind, wobei wir voraussetzen können, dass die linke Seite der Implikation gilt.

$0 = 0(0-1)$	offenbar gültig
$0 \leq 0$	offenbar gültig
$0 \leq n+1$	gilt wegen der Prämisse $n \geq 0$

- $F_2 \Rightarrow F_6$:

  $Inv \wedge i \leq n \wedge t = t_0 \Rightarrow (Inv \wedge 0 \leq t < t_0)\begin{bmatrix}i+1\\i\end{bmatrix}\begin{bmatrix}s+2i\\s\end{bmatrix}$

  $\cdots \Rightarrow s{+}2i = (i{+}1)(i{+}1-1) \wedge 0 \leq i{+}1 \leq n+1 \wedge 0 \leq n+1-(i{+}1) < t_0$

  $s = i(i-1) \wedge 0 \leq i \leq n+1 \wedge i \leq n \wedge n+1-i = t_0$

  $\Rightarrow s{+}2i = (i{+}1)i \wedge 0 \leq i{+}1 \leq n+1 \wedge 0 \leq n-i < t_0$

  Diese Implikation ist gültig, weil:

$s{+}2i = (i{+}1)i$	arithmetisch äquivalent zur Prämisse $s = i(i-1)$
$0 \leq i{+}1$	folgt aus der Prämisse $0 \leq i$
$i{+}1 \leq n+1$	folgt aus der Prämisse $i \leq n$ durch Addition von 1 auf beiden Seiten
$0 \leq n-i$	äquivalent zur Prämisse $i \leq n$
$n-i < t_0$	wegen $t_0 = n+1-i$ äquivalent zu $n-i < n+1-i$ bzw. nach Kürzen zu $0 < 1$, offenbar gültig

- $F_4 \Rightarrow Post$:

  $$Inv \wedge i \not\leq n \Rightarrow s = n^2 + n$$

  $s = i(i-1) \wedge 0 \leq i \leq n+1 \wedge i \leq n+1 \wedge i > n \Rightarrow s = n^2 + n$

  Aus den Prämissen $i > n$ und $i \leq n+1$ folgt $i = n+1$. Damit erhalten wir aus der ersten Prämisse $s = (n+1)n$, was äquivalent zur Konklusion ist. Somit ist die Implikation gültig.

Alternativlösung: Statt zw↑ kann auch die Annotierungsregel zw↓ verwendet werden. Dann ändern sich die ersten Zeilen des annotierten Programms zu den folgenden.

$\{\, Pre\colon n \geq 0 \,\}$
$i \coloneqq 0$**;**
zw↓ $\{\, F_7\colon n \geq 0 \wedge i = 0 \,\}$
$s \coloneqq 0$**;**
zw↓ $\{\, F_8\colon n \geq 0 \wedge i = 0 \wedge s = 0 \,\}$
wht $\{\, F_1\colon Inv \,\}$

Anstelle von $Pre \Rightarrow F_8$ ist dann die Gültigkeit folgender Implikation zu zeigen.

- $F_8 \Rightarrow F_1$:

  $n \geq 0 \wedge i = 0 \wedge s = 0 \Rightarrow Inv$

  $n \geq 0 \wedge i = 0 \wedge s = 0 \Rightarrow s = i(i-1) \wedge 0 \leq i \leq n+1$

  Diese Implikation ist gültig, weil:

$s = i(i-1)$	folgt aus den Prämissen $i = 0$ und $s = 0$
$0 \leq i$	folgt aus der Prämisse $i = 0$
$i \leq n+1$	folgt aus den Prämissen $i =$ und $n \geq 0$