Theoretische Informatik und Logik
Übungsblatt 4 (2020W)
Lösungsvorschlag

Allgemeine Hinweise: Nummerieren Sie alle auftretenden Formeln in Tableau-Beweisen und geben Sie entsprechende Herkunftshinweise bei allen Regelanwendungen an. Außerdem sind γ- und δ-Formeln jeweils als solche zu markieren. Beachten Sie die Notationsvereinbarungen auf Folie 321.

Aufgabe 4.1

Für jede der folgenden Behauptungen: Finden Sie entweder einen Tableau-Beweis oder geben Sie ein Gegenbeispiel an.

a) Aus $\forall x \exists y P(x,y)$ und $\exists x[P(x,x) \supset \forall y P(x,f(y))]$ folgt $\exists x \forall y[P(x,y) \supset x \neq y \vee y = f(x)]$.

b) Aus $\forall x\, f(x) = x$, $\forall x \exists y f(x) = f(g(y))$ und $\exists x\, g(x) = a$ folgt $\exists x[f(x) = a \wedge \exists y\, g(x) = f(y)]$.

c) $\forall z \exists x \forall y[R(x,y,z) \wedge \forall x \exists y \neg R(x,y,z)]$ ist unerfüllbar.

Lösung 4.1

a) Die Konsequenzbehauptung ist falsch, wie folgendes Gegenbeispiel zeigt:
$\mathcal{I} = \langle \omega, \Phi, \xi \rangle$, mit $\Phi(P)(m,n) \Longleftrightarrow m \leq n$, $\Phi(f)(n) = n+1$, ξ ist irrelevant und kann daher beliebig gewählt werden, da hier nur geschlossene Formeln interpretiert werden.

Die Formel $\forall x \exists y P(x,y)$ besagt unter dieser Interpretation, dass es zu jeder natürlichen Zahl eine mindestens so große Zahl gibt, ist also wahr. Die zweite Prämisse drückt unter $\mathcal{I}$ Folgendes aus: Es gibt eine Zahl k, sodass $k \leq k$ impliziert dass $k \leq n$ für alle n. Wenn wir $k = 0$ wählen, wird klar, dass auch diese Formel unter $\mathcal{I}$ wahr ist. Die Konklusion $\exists x \forall y[P(x,y) \supset x \neq y \vee y = f(x)]$ hingegen ist unter $\mathcal{I}$ falsch, denn sie besagt, dass es eine Zahl m gibt, sodass für alle $n \geq m$ entweder $n \neq m$ oder $n = m+1$ (oder beides) gilt.

b) Folgendes geschlossene Tableau zeigt die Richtigkeit der Konsequenzbehauptung:

(1)	$\mathbf{t} : \forall x\, f(x) = x$				Annahme, γ-Formel
(2)	$\mathbf{t} : \forall x \exists y f(x) = f(g(y))$				Annahme, γ-Formel
(3)	$\mathbf{t} : \exists x\, g(x) = a$				Annahme, δ-Formel
(4)	$\mathbf{f} : \exists x[f(x) = a \wedge \exists y\, g(x) = f(y)]$				Annahme, γ-Formel
(5)	$\mathbf{f} : f(a) = a \wedge \exists y\, g(a) = f(y)$				von 4
(6)	$\mathbf{f} : f(a) = a$	von 5	(7)	$\mathbf{f} : \exists y\, g(a) = f(y)$	von 5 (γ-Formel)
(8)	$\mathbf{t} : f(a) = a$	von 1	(9)	$\mathbf{f} : g(a) = f(g(a))$	von 7
	$\times$	(Wid. 6/8)	(10)	$\mathbf{t} : f(g(a)) = g(a)$	von 1
			(11)	$\mathbf{f} : g(a) = g(a)$	$S^{=}$: [10$\rightarrow$ 9]
				$\times$	$AB^{=}$

Beachten Sie, dass es nicht möglich ist den rechten Ast des Tableaus bereits unmittelbar nach Zeile 10 abzuschließen: Es gibt keine Tableau-Regel, die das gestatten würde (siehe Folien 455/456). Durch diesen Tableau-Beweis wird außerdem klar, dass bereits die stärkere Konsequenzbehauptung $\forall x\, f(x) = x \models \exists x[f(x) = a \wedge \exists y\, g(x) = f(y)]$ gilt.

c) Die Behauptung ist richtig, wie folgender Tableau-Beweis zeigt:

(1)	$\mathbf{t} : \forall z \exists x \forall y [R(x,y,z) \wedge \forall x \exists y \neg R(x,y,z)]$	Annahme, γ-Formel
(2)	$\mathbf{t} : \exists x \forall y [R(x,y,a) \wedge \forall x \exists y \neg R(x,y,a)]$	von 1, δ-Formel
(3)	$\mathbf{t} : \forall y [R(b,y,a) \wedge \forall x \exists y \neg R(x,y,a)]$	von 2, γ-Formel
(4)	$\mathbf{t} : R(b,a,a) \wedge \forall x \exists y \neg R(x,y,a)$	von 3
(5)	$\mathbf{t} : R(b,a,a)$	von 4
(6)	$\mathbf{t} : \forall x \exists y \neg R(x,y,a)$	von 4, γ-Formel
(7)	$\mathbf{t} : \exists y \neg R(b,y,a)$	von 6, δ-Formel
(8)	$\mathbf{t} : \neg R(b,c,a)$	von 7
(9)	$\mathbf{t} : R(b,c,a) \wedge \forall x \exists y \neg R(x,y,a)$	von 3
(10)	$\mathbf{t} : R(b,c,a)$	von 9
(11)	$\mathbf{t} : \forall x \exists y \neg R(x,y,a)$	von 9, γ-Formel
(12)	$\mathbf{f} : R(b,c,a)$	von 8
	$\times$	Widerspruch (10/12)

Beachten Sie, dass die δ-Regel jeweils nach einer neuen Konstante verlangt. Entsprechend muss die γ-Regel — nach Verwendung der δ-Regel in den Zeilen 3 und 8 — nochmals auf die γ-Formel in der Zeile 3 angewendet werden, um ein geschlossenes Tableau zu erhalten.

Aufgabe 4.2

Sind folgende Konsequenzbehauptungen richtig? Im positiven Fall ist ein Tableau-Beweis anzugeben, im negativen Fall ein vollständig spezifiziertes Gegenbeispiel. (Siehe Folie 409 für entsprechende PL-Formeln.)

a) Jede partiell funktionale und reflexive Relation ist total funktional.

b) Jede partiell funktionale, symmetrische und euklidische Relation ist total funktional.

Lösung 4.2

a) Ein Tableau-Beweis dieser Konsequenzbehauptung sieht wie folgt aus:

(1)			$\mathbf{t} : \forall x \forall y \forall z [(R(x,y) \wedge R(x,z)) \supset y = z]$			partielle Funktionalität (γ-F.)
(2)			$\mathbf{t} : \forall x R(x,x)$			Reflexivität (γ-Formel)
(3)			$\mathbf{f} : \forall x \exists y [R(x,y) \wedge \forall z (R(x,z) \supset y = z)]$			totale Funktionalität (δ-F.)
(4)			$\mathbf{f} : \exists y [R(a,y) \wedge \forall z (R(a,z) \supset y = z)]$			von 3 (γ-Formel)
(5)			$\mathbf{f} : R(a,a) \wedge \forall z (R(a,z) \supset a = z)$			von 4
(6)			$\mathbf{t} : R(a,a)$			von 2
(7)	$\mathbf{f} : R(a,a)$ v. 5 $\times$ (6/7)	(8)	$\mathbf{f} : \forall z (R(a,z) \supset a = z)$			von 5 (δ-Formel)
		(9)	$\mathbf{f} : R(a,c) \supset a = c$			von 8
		(10)	$\mathbf{t} : R(a,c)$			von 9
		(11)	$\mathbf{f} : a = c$			von 9
		(12)	$\mathbf{t} : \forall y \forall z [(R(a,y) \wedge R(a,z)) \supset y = z]$			von 1 (γ-Formel)
		(13)	$\mathbf{t} : \forall z [(R(a,a) \wedge R(a,z)) \supset a = z]$			von 12 (γ-Formel)
		(14)	$\mathbf{t} : (R(a,a) \wedge R(a,c)) \supset a = c$			von 13
		(15)	$\mathbf{f} : R(a,a) \wedge R(a,c)$ von 14		(16) $\mathbf{t} : a = c$ $\times$ (11/16)	von 14
		(17)	$\mathbf{f} : R(a,a)$ v. 15 $\times$ (6/17)	(18)	$\mathbf{f} : R(a,c)$ v. 15 $\times$ (10/18)	

b) Diese Aussage ist falsch. Ein einfaches Gegenbeispiel besteht in einer ein-elementigen Domäne mit leerer Relation.

Formaler: Wir definieren die Interpretation $\mathcal{I} = \langle \{0\}, \Phi, \xi \rangle$, wobei $\Phi(R)(0,0) = \mathbf{f}$. (Die Variablenbelegung ist in diesem Fall eindeutig, da es nur ein Domänelement gibt; sie ist allerdings irrelevant.)
Es ist leicht zu sehen, dass folgende Formeln unter $\mathcal{I}$ wahr sind:
$\forall x \forall y \forall z [R(x,y) \wedge R(x,z) \supset y = z]$ (partielle Funktionalität)
$\forall x \forall y [R(x,y) \supset R(y,x)]$ (Symmetrie)
$\forall x \forall y \forall z [R(x,y) \wedge R(x,z) \supset R(y,z)]$ (Euklidizität)

Hingegen ist folgende Formel falsch unter der Interpretation $\mathcal{I}$:
$\forall x \exists y[R(x,y) \wedge \forall z(R(x,z) \supset y = z)]$ (totale Funktionalität)
Damit ist gezeigt, dass $\mathcal{I}$ ein Gegenbeispiel zur Behauptung ist, dass aus der partiellen Funktionalität, Symmetrie und Euklidizität einer Relation deren totale Funktionalität folgt.

Aufgabe 4.3

Untersuchen Sie für folgende Regeln eines Frege-Hilbert-Typ-Kalküls, ob sie korrekt sind. Falls ja, zeigen Sie warum; falls nein, geben Sie ein konkretes Gegenbeispiel an.

a) R1: $\dfrac{A \supset B \qquad C \wedge B \supset D}{A \supset (C \supset D)}$

b) R2: $\dfrac{\neg A \wedge (B \vee C) \qquad \neg(A \supset C)}{D}$

c) R3: $\dfrac{A \supset B}{\exists x A \supset B}$

Hinweise:
(1) A, B, C, D stehen für beliebige Formeln. Sie können aber hier von atomaren Formeln ausgehen.
(2) Beachten Sie, dass R3 – im Unterschied zur Regel auf Folie 441 – keine Nebenbedingung hat.

Lösung 4.3

Wie auf Folie vermerkt, ist ein Kalkül korrekt, wenn nur richtige Konsequenzbehauptungen damit bewiesen werden können. Eine Kalkül-Regel ist daher korrekt, wenn Sie einer waren Konsequenzbehauptung entspricht und inkorrekt, wenn es ein Gegenbeispiel zur entsprechenden Konsequenzbehauptung gibt.

Wie schon in den Hinweisen vermerkt, reicht es ohne Beschränkung der Allgemeinheit aus jeweils atomare Instanzen für A, B, C, D zu betrachten.

a) R1 entspricht der Konsequenzbehauptung $A \supset B, C \wedge B \supset D \models A \supset (C \supset D)$.
Diese ist richtig, wie folgender Tableau-Beweis zeigt:

(1)	$\mathbf{t} : A \supset B$	Annahme
(2)	$\mathbf{t} : C \wedge B \supset D$	Annahme
(3)	$\mathbf{f} : A \supset (C \supset D)$	Annahme (negierte Konklusion)
(4)	$\mathbf{t} : A$	von 3
(5)	$\mathbf{f} : C \supset D$	von 3
(6)	$\mathbf{t} : C$	von 5
(7)	$\mathbf{f} : D$	von 5
(8)	$\mathbf{f} : A$ von 1 × (4/8) | (9) $\mathbf{t} : B$	von 1
	(10) $\mathbf{f} : C \wedge B$ von 2 | (11) $\mathbf{t} : D$	von 2
	(12) $\mathbf{f} : C$ von 10 × (6/12) | (13) $\mathbf{f} : B$ von 10 × (9/13) | × (7/11)	

Auch, beispielsweise, ein Nachweis mittels einer Wahrheitstafel, die alle 16 möglichen Interpretationen explizit auflistet, wäre möglich.

b) Die R2 entsprechende Konsequenzbehauptung $\neg A \wedge (B \vee C), \neg(A \supset C) \models D$ ist richtig, wie folgender Tableau-Beweis zeigt:

(1)	$\mathbf{t} : \neg A \wedge (B \vee C)$	Annahme
(2)	$\mathbf{t} : \neg(A \supset C)$	Annahme
(3)	$\mathbf{f} : D$	Annahme (negierte Konklusion)
(4)	$\mathbf{t} : \neg A$	von 1
(5)	$\mathbf{t} : B \vee C$	von 1
(6)	$\mathbf{f} : A \supset C$	von 2
(7)	$\mathbf{t} : A$	von 6
(8)	$\mathbf{f} : C$	von 6
(9)	$\mathbf{f} : A$	von 4
	× (7/9)	

Auch, beispielsweise, folgende informellere Begründung zeigt die Korrektheit von R2:
Die rechte Prämisse $\neg(A \supset C)$ ist äquivalent zu $A \wedge \neg C$. Somit besteht ein Widerspruch zur linken Prämisse, die die Form $\neg A \wedge \ldots$ hat. Es gibt also keine Interpretation die beide Formeln wahr macht. Somit folgt aus diesen beiden Prämissen Beliebiges, also auch D.

c) R3 ist nicht korrekt. Um das zu sehen wählen wir $A = B = P(x)$ und die Interpretation $\mathcal{I} = \langle \omega, \Phi, \xi \rangle$, mit $\Phi(P)(n) \iff n = 0$ und $\xi(x) = 1$. Unter dieser Interpretation besagt die Prämisse der Regel: $1 = 0$ impliziert $1 = 0$ und ist wahr. Aber die Konklusion ist unter $\mathcal{I}$ falsch, denn Sie besagt: Wenn es eine natürlich Zahl gibt, die gleich 0 ist, dann ist auch 1 gleich 0.

Aufgabe 4.4

Sind die folgenden Korrektheitsaussagen wahr oder falsch hinsichtlich partieller bzw. totaler Korrektheit (im Datentyp $\mathbb{Z}$)? Argumentieren Sie informell mit Hilfe der Definition der Semantik von Korrektheitsaussagen. Es ist keine Ableitung im Hoare-Kalkül notwendig.

a) $\{x = x_0 \wedge n > x\}\ x \leftarrow x + n\ \{x > x_0\}$

b) $\{x = x_0 \wedge n > x\}$ `begin` $x \leftarrow x + n$; `while` $x < 0$ `do` $n \leftarrow 0$ `end` $\{x > x_0\}$

c) $\{x = x_0 \wedge n > x\}$ `begin` $x \leftarrow x + n$; `while` $x < 0$ `do` $x \leftarrow -x$ `end` $\{x > x_0\}$

d) $\{x = x_0 \wedge n > x\}$ `begin while` $x < 0$ `do` $n \leftarrow 0$; $x \leftarrow x + n$ `end` $\{x > x_0\}$

e) $\{x = x_0 \wedge n > x\}$ `begin while` $x < 0$ `do` $x \leftarrow -x$; $x \leftarrow x + n$ `end` $\{x > x_0\}$

Lösung 4.4

a) Diese Korrektheitsaussage ist weder partiell noch total korrekt.
Gegenbeispiel: $I(x) = I(x_0) = -1$, $I(n) = 0$. Es gilt:
   - Eingabe erfüllt Vorbedingung: $\mathcal{M}(I, x = x_0 \wedge n > x) = \text{true}$
   - Berechnung des Ergebnisses: $\mathcal{M}(I, x \leftarrow x + n) = I'$, wobei $I'(x) = -1 + 0 = -1$ und $I' \overset{x}{\sim} I$
   - Ergebnis erfüllt Nachbedingung nicht: $\mathcal{M}(I', x > x_0) = (I'(x) > I'(x_0)) = (-1 > -1) = \text{false}$

b) Diese Korrektheitsaussage ist partiell aber nicht total korrekt. Sei I die Variablenbelegung zu Beginn und $I' := \mathcal{M}(I, x \leftarrow x + n)$ jene nach der Zuweisung, somit gilt $I'(x) = I(x) + I(n)$. Wir unterscheiden zwei Fälle.

   $I'(x) \geq 0$**:** Die Schleife terminiert ohne einen einzigen Durchlauf, daher terminiert das Programm mit dem Ergebnis I'. Aus der Fallbedingung $I'(x) = I(x)+I(n) \geq 0$ und der Vorbedingung $n > x$ folgt $I(n) > 0$. Daher erhalten wir $I'(x) = I(x)+I(n) > I(x)$. Wegen $I(x) = I(x_0) = I'(x_0)$ ergibt sich daraus $I'(x) > I'(x_0)$. Somit erfüllt das Programmergebnis die Nachbedingung $x > x_0$.

   $I'(x) < 0$**:** Die Schleife wird ausgeführt und terminiert nicht, da der Wert von x in der Schleife nicht verändert wird.

   Zusammengefasst gilt: Wenn das Programm terminiert, gilt die Nachbedingung (partielle Korrektheit), es terminiert aber nicht für alle Eingaben. Ein Gegenbeispiel zur Termination ist die Variablenbelegung aus Lösung a).

c) Diese Korrektheitsaussage ist partiell und total korrekt. Wir führen dieselbe Fallunterscheidung wie in Lösung b) durch. Der erste Fall ist identisch, wir analysieren den zweiten Fall.

   $I'(x) < 0$**:** Die Schleife wird ausgeführt und terminiert nach einem Durchlauf mit einer Variablenbelegung I'', da $I''(x) = -I'(x) > 0$ gilt. Aus der Fallbedingung $I'(x) = I(x) + I(n) < 0$ und der Vorbedingung $n > x$ folgt $I(x) < 0$, wegen $I(x_0) = I(x)$ und $I(x_0) = I'(x_0) = I''(x_0)$ gilt daher auch $I''(x_0) < 0$. Wegen $I''(x) > 0$ erfüllt I'' die Nachbedingung $x > x_0$.

   Zusammengefasst gilt: Das Programm terminiert immer und das Ergebnis erfüllt die Nachbedingung, somit ist die Korrektheitsaussage wahr hinsichtlich partieller und totaler Korrektheit.

d) Diese Korrektheitsaussage ist partiell aber nicht total korrekt. Sei I die Variablenbelegung zu Beginn. Wir unterscheiden zwei Fälle.

$I(x) \geq 0$**:** Die Schleife terminiert ohne einen einzigen Durchlauf, daher terminiert das Programm mit dem Ergebnis $I' := \mathcal{M}(I, x \leftarrow x + n)$. Aus der Fallbedingung $I(x) \geq 0$ und der Vorbedingung $n > x$ folgt $I(n) > 0$. Daher erhalten wir $I'(x) = I(x) + I(n) > I(x)$. Wegen $I(x) = I(x_0) = I'(x_0)$ ergibt sich daraus $I'(x) > I'(x_0)$. Das heißt, dass das Ergebnis des Programms die Nachbedingung $x > x_0$ erfüllt.

$I(x) < 0$**:** Die Schleife wird ausgeführt und terminiert nicht, da der Wert von x in der Schleife nicht verändert wird.

Zusammengefasst gilt: Wenn das Programm terminiert, gilt die Nachbedingung (partielle Korrektheit), es terminiert aber nicht für alle Eingaben. Ein Gegenbeispiel zur Termination ist die Variablenbelegung aus Lösung a).

e) Diese Korrektheitsaussage ist partiell und total korrekt. Wir führen dieselbe Fallunterscheidung wie in Lösung d) durch. Der erste Fall ist identisch, wir analysieren den zweiten Fall.

$I(x) < 0$**:** Die Schleife wird ausgeführt und terminiert nach einem Durchlauf mit einer Variablenbelegung I', da $I'(x) = -I(x) > 0$ gilt. Nach Ausführung der Zuweisung erhalten wir die Belegung $I'' := \mathcal{M}(I', x \leftarrow x + n)$, es gilt insbesondere $I''(x) = I'(x) + I'(n) = I'(x) + I(n)$.

Für $I(n) \geq 0$ gilt jedenfalls $I''(x) \geq I'(x) > 0$. Aber auch für $I(n) < 0$ ist wegen der Vorbedingung $n > x$ sichergestellt, dass n betragsmäßig kleiner als x ist, somit ist $I''(x)$ zwar kleiner als $I'(x)$, aber immer noch positiv. Da $I(x)$ und damit $I(x_0) = I''(x_0)$ negativ ist, erfüllt I'' die Nachbedingung $x > x_0$.

Zusammengefasst gilt: Das Programm terminiert immer und das Ergebnis erfüllt in beiden Fällen die Nachbedingung, somit ist die Korrektheitsaussage wahr hinsichtlich partieller und totaler Korrektheit.

Aufgabe 4.5

Zeigen Sie mit Hilfe des Hoare-Kalküls, dass die folgende Korrektheitsaussage (im Datentyp $\mathbb{Z}$) wahr hinsichtlich totaler Korrektheit („total korrekt“) ist. Verwenden Sie die Formel $\boldsymbol{l * y \leq x < h * y \wedge y > 0}$ als Invariante und den Ausdruck $\boldsymbol{h - l}$ als Variante. Welche Funktion berechnet das Programm, wenn man l als das Ergebnis des Programms betrachtet?

$\{y > 0 \wedge x \geq 0\}$
```
begin
  begin
    l ← 0;
    h ← x + 1
  end;
  while l + 1 ≠ h do
    begin
      z ← (l + h)/2;
      if z * y > x then
        h ← z
      else
        l ← z
    end
end
```
$\{l * y \leq x < (l + 1) * y\}$

Lösung 4.5

Wir beginnen damit, das Programm entsprechend der Regeln des Hoare-Kalküls zu annotieren. Wir nummerieren die Formeln in der Reihenfolge des Hinzufügens. Die vorgeschlagene Reihenfolge ist nur eine von mehreren Möglichkeiten.

$\{Pre\colon y > 0 \wedge x \geq 0\}$
$\{F_6\colon Inv[^{x}_{h} + 1][^{0}_{l}]\}$
```
begin
  begin
```

$l \leftarrow 0$;
$\{F_5\colon Inv[^{x}_{h}\,{}^{+}\,{}^{1}]\}$
$h \leftarrow x + 1$
end;
$\{F_1\colon Inv\}$
while $l + 1 \neq h$ **do**
$\{F_2\colon Inv \wedge t = t_0 \wedge l + 1 \neq h\}$
begin
$z \leftarrow (l + h)/2$;
$\{F_7\colon Inv \wedge t = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2\}$
if $z * y > x$ **then**
$\{F_8\colon Inv \wedge t = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge z * y > x\}$
$\{F_{13}\colon (Inv \wedge 0 \leq t < t_0)[^{z}_{h}]\}$
$h \leftarrow z$
$\{F_{11}\colon Inv \wedge 0 \leq t < t_0\}$
else
$\{F_9\colon Inv \wedge t = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge \neg(z * y > x)\}$
$\{F_{12}\colon (Inv \wedge 0 \leq t < t_0)[^{z}_{l}]\}$
$l \leftarrow z$
$\{F_{10}\colon Inv \wedge 0 \leq t < t_0\}$
end
$\{F_3\colon Inv \wedge 0 \leq t < t_0\}$
end
$\{F_4\colon Inv \wedge \neg(l + 1 \neq h)\}$
$\{Post\colon l * y \leq x < (l + 1) * y\}$

Wir müssen die Gültigkeit von vier Implikationen zeigen: $Pre \supset F_6$, $F_8 \supset F_{13}$, $F_9 \supset F_{12}$ und $F_4 \supset Post$. Dabei verwenden wir die in der Angabe vorgeschlagene Invariante und Variante:

$$\begin{aligned} Inv &= l * y \leq x < h * y \wedge y > 0 \\ t &= h - l \end{aligned}$$

Bei jeder der vier Implikationen müssen wir zeigen, dass jedes Konjunkt auf der rechten Seite der Implikation aus den Prämissen (Konjunkte der linken Seite der Implikation) sowie den Gesetzen der ganzzahligen Arithmetik folgt.

$$\begin{aligned} & Pre \supset F_6 \\ y > 0 \wedge x \geq 0 &\supset Inv[^{x}_{h}\,{}^{+}\,{}^{1}][^{0}_{l}] \\ y > 0 \wedge x \geq 0 &\supset (l * y \leq x < h * y \wedge y > 0)[^{x}_{h}\,{}^{+}\,{}^{1}][^{0}_{l}] \\ y > 0 \wedge x \geq 0 &\supset (l * y \leq x < (x + 1) * y \wedge y > 0)[^{0}_{l}] \\ y > 0 \wedge x \geq 0 &\supset 0 * y \leq x < (x + 1) * y \wedge y > 0 \end{aligned}$$

Konjunkt	gilt, weil ...
$0 * y \leq x$	äquivalent zu $0 \leq x$ (Prämisse)
$x < (x + 1) * y$	$x < x + 1$ (Arithmetik), $x + 1 \leq (x + 1) * y$ falls $x + 1 \geq 0$ und $y \geq 1$ (Arithmetik), $x + 1 \geq 0$ (Prämisse $x \geq 0$), $y \geq 1$ (Prämisse $y > 0$)
$y > 0$	Prämisse

$$\begin{aligned} & F_8 \supset F_{13} \\ Inv \wedge t = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge z * y > x &\supset (Inv \wedge 0 \leq t < t_0)[^{z}_{h}] \\ l * y \leq x < h * y \wedge y > 0 \wedge h - l = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge z * y > x &\supset (l * y \leq x < h * y \wedge y > 0 \wedge 0 \leq h - l < t_0)[^{z}_{h}] \\ l * y \leq x < h * y \wedge y > 0 \wedge h - l = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge z * y > x &\supset l * y \leq x < z * y \wedge y > 0 \wedge 0 \leq z - l < t_0 \end{aligned}$$

Konjunkt	gilt, weil ...
$l * y \leq x$	Prämisse
$x < z * y$	Prämisse
$y > 0$	Prämisse
$0 \leq z - l$	aus $l * y \leq x$ (Prämisse) und $z * y > x$ (Prämisse) folgt $l * y \leq z * y$, äquivalent zu $l \leq z$ (Division durch y, möglich wegen Prämisse $y > 0$), äquivalent zu $0 \leq z - l$
$z - l < t_0$	äquivalent zu $z - l < h - l$ (Prämisse $h - l = t_0$), äquivalent zu $z < h$ (Arithmetik), äquivalent zu $(l + h)/2 < h$ (Prämisse $z = (l + h)/2$). Diese Ungleichung gilt, weil: Aus $l * y \leq x < h * y$ (Prämisse) folgt $l < h$ (Division durch y möglich wegen Prämisse $y > 0$) bzw. $l + 1 \leq h$ (Arithmetik). Wegen der Prämisse $l + 1 \neq h$ gilt aber sogar $l + 2 \leq h$ bzw. $l \leq h - 2$. Daraus erhalten wir $l + h \leq 2 * h - 2$ (Addition von h), äquivalent zu $(l + h)/2 \leq h - 1$, was gleichbedeutend mit $(l + h)/2 < h$ ist.

$$F_9 \supset F_{12}$$

$$Inv \wedge t = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge \neg(z * y > x) \supset (Inv \wedge 0 \leq t < t_0)[^z_l]$$

$$l * y \leq x < h * y \wedge y > 0 \wedge h - l = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge \neg(z * y > x) \supset (l * y \leq x < h * y \wedge y > 0 \wedge 0 \leq h - l < t_0)[^z_l]$$

$$l * y \leq x < h * y \wedge y > 0 \wedge h - l = t_0 \wedge l + 1 \neq h \wedge z = (l + h)/2 \wedge z * y \leq x \supset z * y \leq x < h * y \wedge y > 0 \wedge 0 \leq h - z < t_0$$

Konjunkt	gilt, weil ...
$z * y \leq x$	Prämisse
$x < h * y$	Prämisse
$y > 0$	Prämisse
$0 \leq h - z$	aus $x < h * y$ (Prämisse) und $z * y \leq x$ (Prämisse) folgt $z * y \leq h * y$, äquivalent zu $z \leq h$ (Division durch y, möglich wegen Prämisse $y > 0$), äquivalent zu $0 \leq h - z$
$h - z < t_0$	äquivalent zu $h - z < h - l$ (Prämisse $h - l = t_0$), äquivalent zu $l < z$ (Arithmetik), äquivalent zu $l < (l + h)/2$ (Prämisse $z = (l + h)/2$). Diese Ungleichung gilt, weil: Aus $l * y \leq x < h * y$ (Prämisse) folgt $l < h$ (Division durch y möglich wegen Prämisse $y > 0$) bzw. $l + 1 \leq h$ (Arithmetik). Wegen der Prämisse $l + 1 \neq h$ gilt aber sogar $l + 2 \leq h$. Daraus erhalten wir $2 * l + 2 \leq l + h$ (Addition von l), äquivalent zu $l + 1 \leq (l + h)/2$, was gleichbedeutend mit $l < (l + h)/2$ ist.

$$F_4 \supset Post$$

$$Inv \wedge \neg(l + 1 \neq h) \supset l * y \leq x < (l + 1) * y$$

$$l * y \leq x < h * y \wedge y > 0 \wedge l + 1 = h \supset l * y \leq x < (l + 1) * y$$

Konjunkt	gilt, weil ...
$l * y \leq x$	Prämisse
$x < (l + 1) * y$	folgt aus $x < h * y$ (Prämisse) und $l + 1 = h$ (Prämisse)

Welche Funktion wird durch das Programm berechnet? Zur Beantwortung dieser Frage reicht es, die Nachbedingung so umzuformen, dass wir das Ergebnis l als Funktion der anderen Variablen erhalten.

$l * y \leq x < (l + 1) * y$	Division durch y
$l \leq x/y < l + 1$	Definition der Floor-Funktion
$l = \lfloor x/y \rfloor$	

Das Programm berechnet also die ganzzahlige Division, l ist das Ergebnis der Division von x durch y.

Schlussbemerkung: Vermutlich ist nach dieser Aufgabe klar, warum die Automatisierung der vielen kleinen „Beweise“ notwendig ist, wenn diese Art der formalen Verifikation für reale Programme verwendet werden soll. Es entstehen schnell Tausende von Formeln, deren Gültigkeit gezeigt werden muss. Heutige Verifikationstool haben eine Erfolgsquote von 99 % und darüber, sodass dem menschlichen Verifikator nur eine Handvoll von Formeln bleiben (was mühsam genug sein kann).