Denotationale Semantik

Ein Beispiel:

<u>in</u> (X,Y);
<u>while</u> X≠0 <u>do</u> X:=X-Y; Y:=Y-X <u>od</u>;
<u>out</u> (Y)

Aufgabe: Berechne speziell die Bedeutung von
<u>while</u> b <u>do</u> c <u>od</u> mit b ≡ X≠0
und c ≡ X:=X-Y; Y:=Y-X

Grundsätzliches Vorgehen:

- ein paar Beispiele durchrechnen,
- die Funktion des Rumpfes der while-Schleife berechnen,
- Eigenschaften über sie notieren (sofern möglich),
- f_1, f_2, f_3 möglichst formal berechnen,
- endlichen Ausschnitt aufschreiben, um etwas zu „erkennen“,
- Hypothesen über die realisierte Funktion aufstellen
- und Beweise führen.

Eingabe (0,1)	Ausgabe 1
Eingabe (0,a)	Ausgabe a
Eingabe (2,2)	Ausgabe 2
Eingabe (a,a)	Ausgabe a
Eingabe (3,2)	Ausgabe 1
Eingabe (6,4)	Ausgabe 2
Eingabe (8,5)	Ausgabe 1

Terminiert die while-Schleife immer?

Leider nein, z.B. nicht bei Eingabe (1,0).
Man probiert aus und stellt fest, dass die while-Schleife für die meisten Werte zu divergieren scheint.

3

$\mathbf{f_c}$=? X:=X-Y; Y:=Y-X liefert: $\mathbf{f_c}(a,b) = (a-b, 2b-a)$

Bemerkung: $\mathbf{f_c}$ ist injektiv,
d.h., aus $\mathbf{f_c}(a,b) = \mathbf{f_c}(x,y)$ folgt $a = x$ und $b = y$.

Der Beweis ist einfach,
denn $\mathbf{f_c}(a,b) = \mathbf{f_c}(x,y)$ bedeutet $(a-b,2b-a) = (x-y,2y-x)$, also
$a-b = x-y$ und $2b-a = 2y-x$, d.h., $b + (b-a) = y + (y-x)$.
Einsetzen ergibt $b = y$ und $a = x$.

4

$f_c(a,b) = (a\text{-}b, 2b\text{-}a)$ als unendlich große Funktionstabelle:

X = \ Y=	…	-3	-2	-1	0	1	2	3	4	5	6	7	8	…
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
-3	…	0,-3	-1,-1	-2,1	-3,3	-4,5	-5,7	-6,9	-7,11	-8,13	-9,15	-10,17	-11,19	…
-2	…	1,-4	0,-2	-1,0	-2,2	-3,4	-4,6	-5,8	-6,10	-7,12	-8,14	-9,16	-10,18	…
-1	…	2,-5	1,-3	0,-1	-1,1	-2,3	-3,5	-4,7	-5,9	-6,11	-7,13	-8,15	-9,17	…
0	…	3,-6	2,-4	1,-2	0,0	-1,2	-2,4	-3,6	-4,8	-5,10	-6,12	-7,14	-8,16	…
1	…	4,-7	3,-5	2,-3	1,-1	0,1	-1,3	-2,5	-3,7	-4,9	-5,11	-6,13	-7,15	…
2	…	5,-8	4,-6	3,-4	2,-2	1,0	0,2	-1,4	-2,6	-3,8	-4,10	-5,12	-6,14	…
3	…	6,-9	5,-7	4,-5	3,-3	2,-1	1,1	0,3	-1,5	-2,7	-3,9	-4,11	-5,13	…
4	…	7,-10	6,-8	5,-6	4,-4	3,-2	2,0	1,2	0,4	-1,6	-2,8	-3,10	-4,12	…
5	…	8,-11	7,-9	6,-7	5,-5	4,-3	3,-1	2,1	1,3	0,5	-1,7	-2,9	-3,11	…
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Betrachte nun: $\Gamma_{b,c}(f) = \textbf{cond}\,(X\neq 0, f\circ f_c, id)$

Hinweis zur formalen Korrektheit:
X bedeutet die Projektion auf die erste Komponente;
id ist hier die Identität auf $\mathbb{Z}\times\mathbb{Z}$, also auf der Menge der Paare ganzer Zahlen.

Es gilt also für alle ganzen Zahlen a und b:

$$\Gamma_{b,c}(f)(a,b) = \textbf{cond}\,(X\neq 0, f\circ f_c, id)(a,b)$$

$$= \begin{cases} f(a-b, 2b-a) & \text{, falls } a\neq 0 \\ (a,b) & \text{, falls } a=0 \end{cases}$$

Hinweis zur formalen Korrektheit: $\Gamma_{b,c}(f)$ ordnet der Speicherbelegung
$X \to a$, $Y \to b$ für $a \neq 0$ die Speicherbelegung $X \to a\text{-}b$, $Y \to 2b\text{-}a$
und anderenfalls unverändert die Speicherbelegung $X \to a$, $Y \to b$ zu.

Man setze gemäß der denotationalen Semantik:

$\mathbf{f_0 = f_\perp}$ (= die überall undefinierte Funktion $f_\perp(a,b) = (\perp,\perp)$ für alle a und b)

$\mathbf{f_{i+1} = \Gamma_{b,c}(f_i)}$ **für alle** $\mathbf{i \geq 0}$

und berechne von $\Gamma_{b,c}$ den kleinsten Fixpunkt $\mathbf{fix(\Gamma_{b,c})}$, also

$$\mathbf{fix}\,(\Gamma_{b,c}) = \lim_{i \to \infty} \mathbf{f_i}\ .$$

Dies ist die Bedeutung der while-Schleife.

Für unser Beispiel erhalten wir:

$\mathbf{f_0 = f_\perp}$ **(= die überall undefinierte Funktion $f_\perp(a,b) = (\perp,\perp)$ für alle a und b)**

$\mathbf{f_1 = \Gamma_{b,c}(f_0) = cond\ (X{\neq}0, f_0 \circ f_c, id)}$**, d.h.** $\forall a,b \in$ 9 **:**

$$\mathbf{f_1(a,b)} = \begin{cases} \mathbf{f_0(a-b,2b-a) = (\perp,\perp)} & \mathbf{, falls\ a \neq 0} \\ \mathbf{(a,b)} & \mathbf{, falls\ a = 0} \end{cases}$$

$\mathbf{f_2 = \Gamma_{b,c}(f_1) = cond\ (X{\neq}0, f_1 \circ f_c, id)}$**, d.h.** $\forall a,b \in$ 9 **:**

$$\mathbf{f_2(a,b)} = \begin{cases} \mathbf{f_1(a-b,2b-a)} & \mathbf{, falls\ a \neq 0} \\ \mathbf{(a,b)} & \mathbf{, falls\ a = 0} \end{cases}$$

$$= \begin{cases} \mathbf{f_1(a-b,2b-a) = (\perp,\perp)} & \mathbf{, falls\ a-b \neq 0} \\ \mathbf{(a-b,2b-a)} & \mathbf{, falls\ a-b = 0} \end{cases} \Bigg\}\ \mathbf{und\ a \neq 0} \\ \quad \mathbf{(a,b)} \qquad\qquad\qquad\qquad \mathbf{, falls\ a = 0}$$

$$f_3 = \Gamma_{b,c}(f_2) = \mathrm{cond}\,(X \neq 0, f_2 \circ f_c, id), \text{ d.h. } \forall a,b \in 9:$$

$$f_3(a,b) = \begin{cases} f_2(a-b, 2b-a) & \text{, falls } a \neq 0 \\ (a,b) & \text{, falls } a = 0 \end{cases}$$

$$= \begin{cases} f_1(2a-3b, 5b-3a) & \text{, falls } 2a-3b \neq 0,\ a-b \neq 0,\ a \neq 0 \\ (2a-3b, 5b-3a) & \text{, falls } 2a-3b = 0,\ a-b \neq 0,\ a \neq 0 \\ (a-b, 2b-a) & \text{, falls } a-b = 0 \text{ und } a \neq 0 \\ (a,b) & \text{, falls } a = 0 \end{cases}$$

$$= \begin{cases} (\perp, \perp) & \text{, falls } 2a-3b \neq 0,\ a-b \neq 0,\ a \neq 0 \\ (0, 5b-3a) & \text{, falls } 2a-3b = 0,\ a-b \neq 0,\ a \neq 0 \\ (0, 2b-a) & \text{, falls } a-b = 0 \text{ und } a \neq 0 \\ (0, b) & \text{, falls } a = 0 \end{cases}$$

9

Man rechnet weiter nach:

$$f_4(a,b) = \begin{cases} f_3(a-b, 2b-a) & \text{, falls } a \neq 0 \\ (a,b) & \text{, falls } a = 0 \end{cases}$$

$$= \begin{cases} (\perp, \perp) & \text{, falls } 5a-8b \neq 0,\ 2a-3b \neq 0,\ a-b \neq 0,\ a \neq 0 \\ (0, 13b-8a) & \text{, falls } 5a-8b = 0,\ 2a-3b \neq 0,\ a-b \neq 0,\ a \neq 0 \\ (0, 5b-3a) & \text{, falls } 2a-3b = 0,\ a-b \neq 0,\ a \neq 0 \\ (0, 2b-a) & \text{, falls } a-b = 0 \text{ und } a \neq 0 \\ (0, b) & \text{, falls } a = 0 \end{cases}$$

usw.

„Sieht“ man etwas? Eine Regelmäßigkeit? Die Koeffizienten scheinen aufeinander folgende Fibonaccizahlen zu sein?!

10

$f_0 = f_\perp$ in (X,Y); while X≠0 do X:=X-Y; Y:=Y-X od; out (Y)

X= \ Y=	…	-3	-2	-1	0	1	2	3	4	5	6	7	8	…
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
-3	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-2	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-1	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
0	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
1	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
2	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
3	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
4	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
5	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

11

$f_1 = \Gamma_{b,c}(f_0)$ in (X,Y); while X≠0 do X:=X-Y; Y:=Y-X od; out (Y)

X= \ Y=	…	-3	-2	-1	0	1	2	3	4	5	6	7	8	…
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
-3	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-2	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-1	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
0	…	0,-3	0,-2	0,-1	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	…
1	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
2	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
3	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
4	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
5	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

12

$f_2 = \Gamma_{b,c}(f_1)$ <u>in</u> (X,Y); <u>while</u> X≠0 <u>do</u> X:=X-Y; Y:=Y-X <u>od</u>; <u>out</u> (Y)

X = \ Y =	…	-3	-2	-1	0	1	2	3	4	5	6	7	8	…
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
-3	…	0,-3	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-2	…	⊥,⊥	0,-2	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-1	…	⊥,⊥	⊥,⊥	0,-1	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
0	…	0,-3	0,-2	0,-1	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	…
1	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,1	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
2	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,2	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
3	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,3	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
4	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,4	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
5	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,5	⊥,⊥	⊥,⊥	⊥,⊥	…
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

13

$f_3 = \Gamma_{b,c}(f_2)$ in (X,Y); while X≠0 do X:=X-Y; Y:=Y-X od; out (Y)

X = \ Y =	…	-3	-2	-1	0	1	2	3	4	5	6	7	8	…
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
-3	…	0,-3	**0,-1**	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-2	…	⊥,⊥	0,-2	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-1	…	⊥,⊥	⊥,⊥	0,-1	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
0	…	0,-3	0,-2	0,-1	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	…
1	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,1	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
2	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,2	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
3	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	**0,1**	0,3	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
4	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,4	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
5	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,5	⊥,⊥	⊥,⊥	⊥,⊥	…
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

14

fix ($\Gamma_{b,c}$) <u>in</u> (X,Y); <u>while</u> X≠0 <u>do</u> X:=X-Y; Y:=Y-X <u>od</u>; <u>out</u> (Y)

X = \ Y =	…	-3	-2	-1	0	1	2	3	4	5	6	7	8	…
		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
-3	…	0,-3	0,-1	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-2	…	⊥,⊥	0,-2	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
-1	…	⊥,⊥	⊥,⊥	0,-1	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
0	…	0,-3	0,-2	0,-1	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	…
1	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,1	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
2	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,2	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
3	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,1	0,3	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
4	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,4	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	…
5	…	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	⊥,⊥	0,5	⊥,⊥	⊥,⊥	⊥,⊥	…
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

15

Hypothesen aufstellen (nach einigem Probieren)

Betrachte die Fibonaccizahlen:
$F_0 = 0$, $F_1 = 1$ und für alle $i \geq 0$: $F_{i+2} = F_{i+1} + F_i$.

$F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$, $F_5 = 5$, $F_6 = 8$,
$F_7 = 13$, $F_8 = 21$, $F_9 = 34$, $F_{10} = 55$, $F_{11} = 89$, $F_{12} = 144$, ...

Die while-Schleife terminiert zum Beispiel für die Eingaben (0,1) und (1,1) und (F_{2k} , F_{2k-1}) und ($-F_{2k}$, $-F_{2k-1}$) für alle natürlichen Zahlen k. Auch für deren Vielfache. Aber andere Beispiele findet man nicht ??

16

Setzt man $\mathbf{F_{-1}} = -1$, dann erhält man folgende Hypothese:

Das Programm terminiert genau für alle Eingaben
$$(\mathbf{a \cdot F_{2k}, a \cdot F_{2k-1}})$$
für alle ganzen Zahlen a und alle natürlichen Zahlen k ≥ 0 und liefert dann als Ausgabe die Zahl a.

Selbst versuchen, dies zu beweisen.
Dass das Programm hierfür terminiert, sieht man leicht ein; dass die while-Schleife nur für diese Eingaben terminiert, erkennt man aus der Folge $(X,Y) \to (X-Y, 2Y-X)$.

17

Hinweis: Wenn die while-Schleife für $(X,Y) = (\mathbf{a \cdot F_{2k}, a \cdot F_{2k-1}})$ terminiert und das Ergebnis (0,a) liefert, dann betrachte die Anfangswerte $(X,Y) = (\mathbf{a \cdot F_{2k+2}, a \cdot F_{2k+1}})$. Nach einem Schleifendurchlauf hat sich diese Variablenbelegung dann geändert zu (beachte: $F_{2k+2} - F_{2k+1} = F_{2k}$):

$$\begin{aligned}(X-Y, 2Y-X) &= (a \cdot F_{2k+2} - a \cdot F_{2k+1}, 2a \cdot F_{2k+1} - a \cdot F_{2k+2}) \\ &= (a \cdot F_{2k}, a \cdot F_{2k+1} - a \cdot (F_{2k+2} - F_{2k+1})) \\ &= (a \cdot F_{2k}, a \cdot F_{2k+1} - a \cdot F_{2k}) \\ &= (a \cdot F_{2k}, a \cdot F_{2k-1}),\end{aligned}$$

d.h., auch mit diesen Eingabewerten terminiert die while-Schleife und liefert ebenfalls das Ergebnis (0,a).

Alle anderen Eingaben divergieren gegen $\pm(\infty,-\infty)$.

18