Satz Die Klasse der LB Fkt. ist genau die Klasse der pr. Fkt.

Bew. $\Leftarrow$ ✓

$\Rightarrow$ strukturelle Induktion

Sei $f: \mathbb{N}^k \to \mathbb{N}$ LB durch Programm P.

Seien die in dem LOOP-Prog. vorkommenden Var. $x_0, \dots, x_m$, $m \geq k$.

Mit struktureller Induktion konstr. wir die pr. Fkt $g_P(c^m(a_0, \dots, a_m)) = c^m(b_0, \dots, b_m)$, wobei $a_0, \dots, a_m$ die Werte vor der Ausführg von P sein, $b_0, \dots, b_m$ die Werte nach der Ausführung von P sein.

- Falls $P \equiv x_j := x_i \pm c$, dann konstr.

$$g_P(n) = c^m(d_0(n), \dots, d_{j-1}(n), d_i(n) \pm c, d_{j+1}(n), \dots, d_m(n))$$

– Falls $P \equiv \underline{Q}; \underline{R}$, dann g. St g_Q und g_P:

$$g_P(n) = \underline{g_R}(\underline{g_Q}(n))$$

– Falls $P \equiv \text{LOOP } x_i \text{ DO } \underline{Q} \text{ END}$, dann g. St g_Q:

$$h(0, x) = x$$
$$h(n+1, x) = g_Q(h(n, x))$$
$$g_P(n) = h(d_i(n), n)$$

Dann ist die gewünschte Fkt:

$$f(n_1, \ldots, n_k) = \underbrace{d_0(g_p(0, n_1, \ldots, n_k, \underbrace{0, \ldots, 0}_{m-k}))}$$

$\square$

<u>Def</u> (μ- Operator)

Gegeben $f: \mathbb{N}^{k+1} \to \mathbb{N}$. Dann ist $\mu(f): \mathbb{N}^k \to \mathbb{N}$ wie folgt def.:

$$\mu(f)(n_1, \ldots, n_k) = \min \left\{ n \;\middle|\; \begin{array}{l} f(n, n_1, \ldots, n_k) = 0 \text{ und} \\ \text{für alle } m \leq n \text{ ist} \\ f(m, n_1, \ldots, n_k) \text{ definiert} \end{array} \right\}$$

wobei $\min \emptyset$ ist undefiniert.

<u>Bem</u>: Damit können wir partielle Fkt. def.

<u>Bsp</u> $f(x,y) = 1 \quad \forall x,y \quad \mu(f)(y)$

z. B. $y=1$ undef! Für alle y ist das undef

d.h. $\mu(f) = \Omega$.

Def. Die Klasse der <u>μ-rek. Funktionen</u> ist die kleinste Klasse von Fkt., die die Basisfkt. enthält und abgeschlossen unter Komposition, primitive Rekursion und <u>Anwendung des μ-Operators</u>.

Satz Die Klasse der WB Fkt. ist genau die Klasse der μ-rekursiven Fkt.

Bew: Erweiterung des Beweises zum letzten Satz.

$\Rightarrow$: Sei P ein WHILE-Prog.: "WHILE $x_i \neq 0$ DO Q END"

Es gibt also g_Q als μ-rek. Fkt.

Wie oben: $h(n, x)$ beschreibt die Var-Werte nach dem n-ten Schleifendurchlauf

$$h(0, x) = x$$

$$h(n+1, x) = g_Q(h(n, x))$$

$$h_i(n, x) = d_i(h(n, x))$$

$h_i(n, x) = 0$ gdw. $x_i = 0$ im WHILE-Prog nach dem n-ten Durchlauf

Dann ist <u>$g_P(x) = h(\mu(h_i)(x), x)$</u>.

$\Leftarrow$ Sei g mit Hilfe des μ-Operators def.: $g = \mu(f)$.
Nach Ind. vor. ex WHILE-Prog P_f um f zu berechnen. Dann kann g wie folgt berechnet werden.

$x_0 := 0;$
$y := f(\underline{0}, x_1, \ldots, x_k);$
WHILE $y \neq 0$ DO
  $x_0 := x_0 + 1;$
  $y := f(x_0, x_1, \ldots, x_k)$
END

$\square$

<u>Satz</u> (Kleene)

Für jede n-stellige μ-rek. Fkt f gibt es zwei $n+1$-stellige p.r. Fkt p und q, so dass

$$f(x_1, \ldots, x_k) = \underline{p(x_1, \ldots, x_k, \mu(q)(x_1, \ldots, x_k))}$$

<u>Bew-Skizze</u>: über übersetzung in WHILE-Prog, $X(F)$, zurück übers. $\square$

2.5 Busy-Beaver-Funktion

Totale, wohl-definierte und unberechenbare Fkt.

<u>Def.</u> Eine <u>Beaver-TM der Größe n</u> ist eine det. TM $M = (Z, \Sigma, \Gamma, \delta, z_0, \sqcup, E)$

$Z = \{z_0, \ldots, z_n\}$, $\Sigma = \underline{\{1\}}$, $\Gamma = \{1, \sqcup\}$,

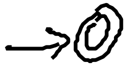
$E = \{z_n\}$, $\delta : (Z - \underline{\{z_n\}}) \times \Gamma \to Z \times \Gamma \times \underline{\{L, R\}}$

Der Score einer Beaver-TM ist die Anzahl der 1en, die auf dem Band stehen, nachdem die TM gestartet auf dem leeren Band anhält.

<u>Def</u>

Eine Beaver-TM der Größe n heißt <u>Busy Beaver</u>, falls sie den maximalen Score unter allen Beaver-TMs der Größe n hat.

Größe 0

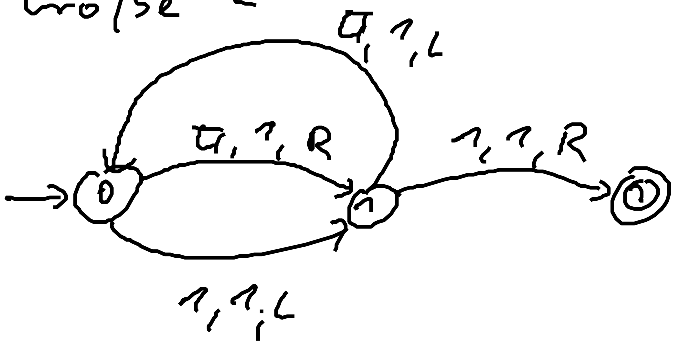


Max Score ist = 0

Größe 1

Max Score ist = 1

Größe 2



1 1 1 1 ist der Busy Beaver

n	#BTM? $(4\times(n+1))^{2n}$	$BB(n)$	
0	1	0	
1	64	1	
2	20736	4	
3	$16{,}8\times10^{6}$	6	
4	$25{,}6\times10^{9}$	13	
5	$63{,}4\times10^{12}$	≥ 4098	
6	232×10^{15}	$\geq 3{,}5\times10^{18267}$	

$BB : \mathbb{N} \to \mathbb{N}$

$BB(n)$ = Score der Busy-Beaver-TM der Größe n ist.

Satz BB ist nicht berechenbar.

Bew: Wir nehmen an, dass BB berechenbar sei mit einer Beaver-TM M_{BB} mit $\boxed{q \text{ Zuständen}}$.

BB ist offensichtlich strikt monoton, d.h.

$x > y \Rightarrow BB(x) > BB(y)$, da mit einem zusätzlichen Zustand mind. eine 1 mehr geschrieben werden kann.

Daraus folgt: $BB(x) > BB(y) \Rightarrow x > y$.

Konstr. Beaver TMs:

1) Für ein bel. gerades n, schreibe $n/2$ 1en aufs Band und fahre zum linken Rand $M_{n/2}$, hat $n/2 + 1$ Zustände (+ Endzustände).

2) Verdopple 1en auf dem Band M_{*2}, hat 6 Zustände (+1 Endzustand)

Schalte zusammen:

$$\boxed{\rightarrow M_{n/2} \rightarrow M_{*2} \rightarrow M_{BB} \Rightarrow}$$

Dies Beaver TM hat $n/2 + 1 + 6 + q$ Zustände

Diese TM schreibt BB(n) aufs leere Band.

$$\Longrightarrow \quad BB(n/2 + 7 + q) \geq BB(n)$$

Mit strikter Monotonie: $BB(n/2 + 8 + q) > BB(n)$

$$n/2 + 8 + q > n \Rightarrow 8 + q > \frac{n}{2} \Rightarrow 16 + 2q > n$$

Das gilt nicht für $n \geq 16 + 2q$. Widerspruch! $\square$

BB wächst stärker als jede berechenbare Fkt.