Wyjątkowo duże liczby

Michał Skrzypczak

Uniwersytet Warszawski

Wydział
Matematyki, Informatyki i Mechaniki

Powered by BeamerikZ

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste:

$0 \in \mathbb{N}$,

$\forall n.\ n \in \mathbb{N} \implies n{+}1 \in \mathbb{N} \qquad \blacksquare$

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste:

$$0 \in \mathbb{N},$$

$$\forall n.\ n \in \mathbb{N} \Longrightarrow n{+}1 \in \mathbb{N} \qquad \blacksquare$$

M2: Liczby naturalne są dane od Boga! (Leopold Kronecker)

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste:

$$0 \in \mathbb{N},$$

$$\forall n.\ n \in \mathbb{N} \implies n{+}1 \in \mathbb{N} \qquad \blacksquare$$

M2: Liczby naturalne są dane od Boga! (Leopold Kronecker)

M3: "*Największa liczba naturalna*" — co za **absurd**!!!

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste:

$$0 \in \mathbb{N},$$

$$\forall n.\ n \in \mathbb{N} \Longrightarrow n{+}1 \in \mathbb{N} \qquad \blacksquare$$

M2: Liczby naturalne są dane od Boga! (Leopold Kronecker)

M3: "*Największa liczba naturalna*" — co za **absurd**!!!

F: Czyli liczba Grahama $G_{64} \in \mathbb{N}$ też jest liczbą naturalną?

(podająca szacowanie w pewnym wariancie twierdzenia Ramseya)

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste:

$0 \in \mathbb{N}$,

$\forall n.\ n \in \mathbb{N} \Longrightarrow n{+}1 \in \mathbb{N} \qquad \blacksquare$

M2: Liczby naturalne są dane od Boga! (Leopold Kronecker)

M3: "*Największa liczba naturalna*" — co za **absurd**!!!

F: Czyli liczba Grahama $G_{64} \in \mathbb{N}$ też jest liczbą naturalną?

(podająca szacowanie w pewnym wariancie twierdzenia Ramseya)

M1, M2, M3 (unisono): Oczywiście!

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste:

$$0 \in \mathbb{N},$$

$$\forall n.\ n \in \mathbb{N} \implies n{+}1 \in \mathbb{N} \qquad \blacksquare$$

M2: Liczby naturalne są dane od Boga! (Leopold Kronecker)

M3: "*Największa liczba naturalna*" — co za **absurd**!!!

F: Czyli liczba Grahama $G_{64} \in \mathbb{N}$ też jest liczbą naturalną?

(podająca szacowanie w pewnym wariancie twierdzenia Ramseya)

M1, M2, M3 (unisono): Oczywiście!

F: Tylko że cząstek w obserwowalnym wszechświecie jest $\sim 10^{80}\ldots$

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste:

$$0 \in \mathbb{N},$$

$$\forall n.\ n \in \mathbb{N} \implies n+1 \in \mathbb{N} \qquad \blacksquare$$

M2: Liczby naturalne są dane od Boga! (Leopold Kronecker)

M3: "*Największa liczba naturalna*" — co za **absurd**!!!

F: Czyli liczba Grahama $G_{64} \in \mathbb{N}$ też jest liczbą naturalną?

(podająca szacowanie w pewnym wariancie twierdzenia Ramseya)

M1, M2, M3 (unisono): Oczywiście!

F: Tylko że cząstek w obserwowalnym wszechświecie jest $\sim 10^{80}$...

$$G_{64} \gg \left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} 10^{80} \text{ razy}$$

Czy liczb naturalnych $n \in \mathbb{N}$ jest nieskończenie wiele?

M1: To proste: „ultrafinityzm”

$$0 \in \mathbb{N},$$

$$\forall n.\ n \in \mathbb{N} \implies n{+}1 \in \mathbb{N} \qquad \blacksquare$$

M2: Liczby naturalne są dane od Boga! (Leopold Kronecker)

M3: “*Największa liczba naturalna*” — co za **absurd**!!!

F: Czyli liczba Grahama $G_{64} \in \mathbb{N}$ też jest liczbą naturalną?

(podająca szacowanie w pewnym wariancie twierdzenia Ramseya)

M1, M2, M3 (unisono): Oczywiście!

F: Tylko że cząstek w obserwowalnym wszechświecie jest $\sim 10^{80}\ldots$

$$G_{64} \gg \left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} 10^{80} \text{ razy}$$

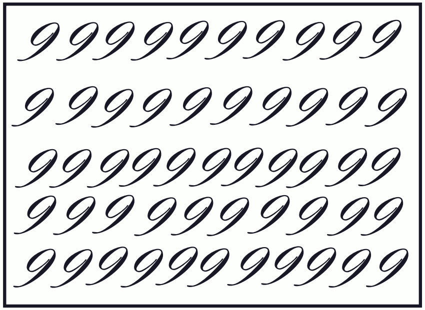
“Who Can Name the Bigger Number?”

“Who Can Name the Bigger Number?” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html

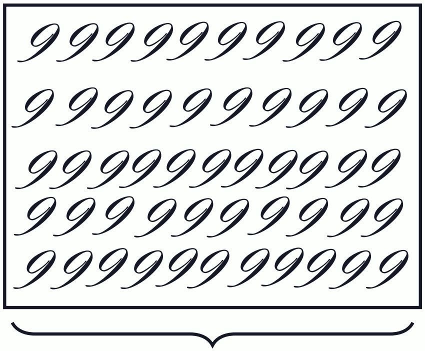
“**Who Can Name the Bigger Number?**” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html



“**Who Can Name the Bigger Number?**” (Scott Aaronson)

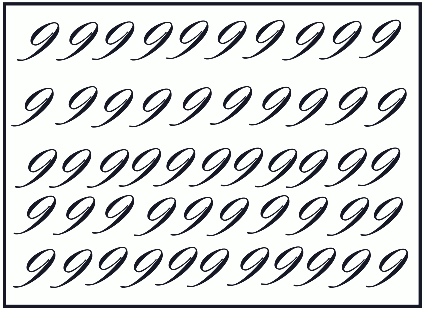
https://www.scottaaronson.com/writings/bignumbers.html



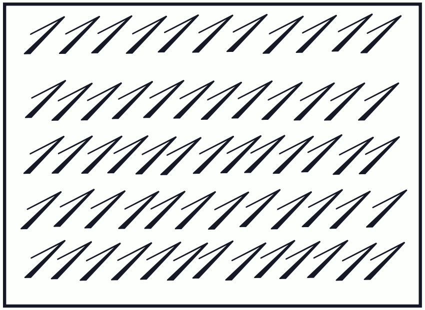
$10^{52} - 1$

“Who Can Name the Bigger Number?” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html

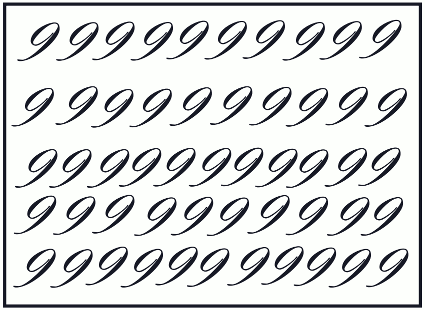


$10^{52} - 1$

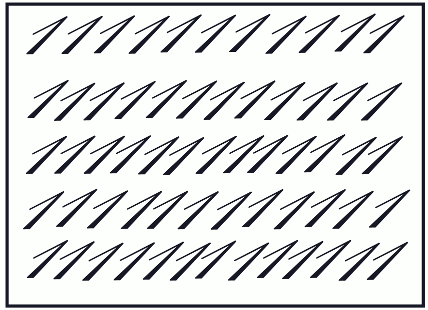


“Who Can Name the Bigger Number?” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html



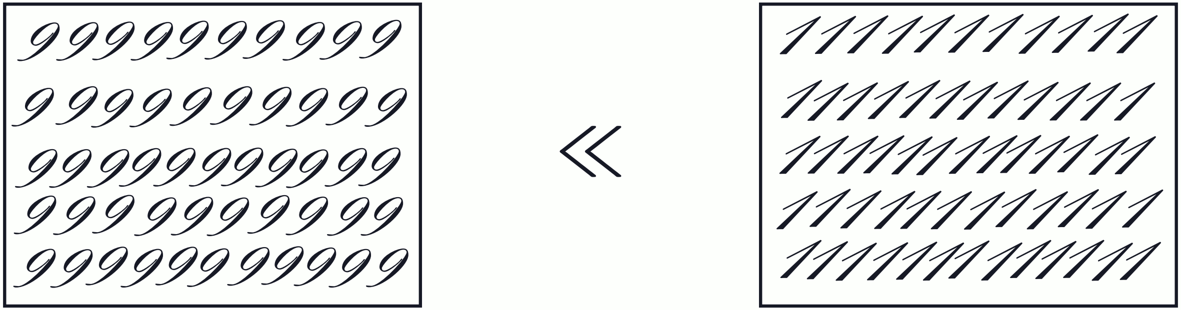
$10^{52} - 1$



$\sim \frac{1}{9} \cdot 10^{61}$

“**Who Can Name the Bigger Number?**” (Scott Aaronson)

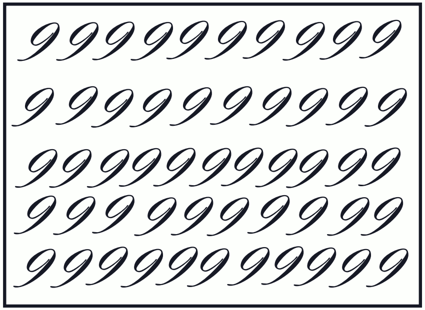
https://www.scottaaronson.com/writings/bignumbers.html



$$\underbrace{}_{10^{52}-1} \quad \ll \quad \underbrace{}_{\sim\frac{1}{9}\cdot 10^{61}}$$

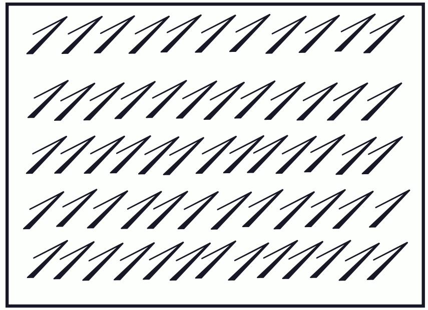
“Who Can Name the Bigger Number?” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html



$10^{52} - 1$

$\ll$

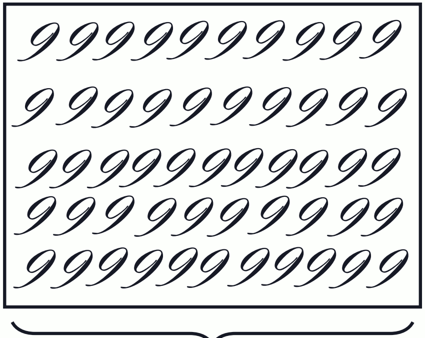


$\sim \frac{1}{9} \cdot 10^{61}$

99999^99999

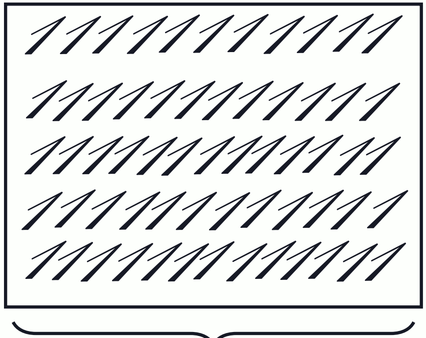
"**Who Can Name the Bigger Number?**" (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html



$10^{52} - 1$

$\ll$



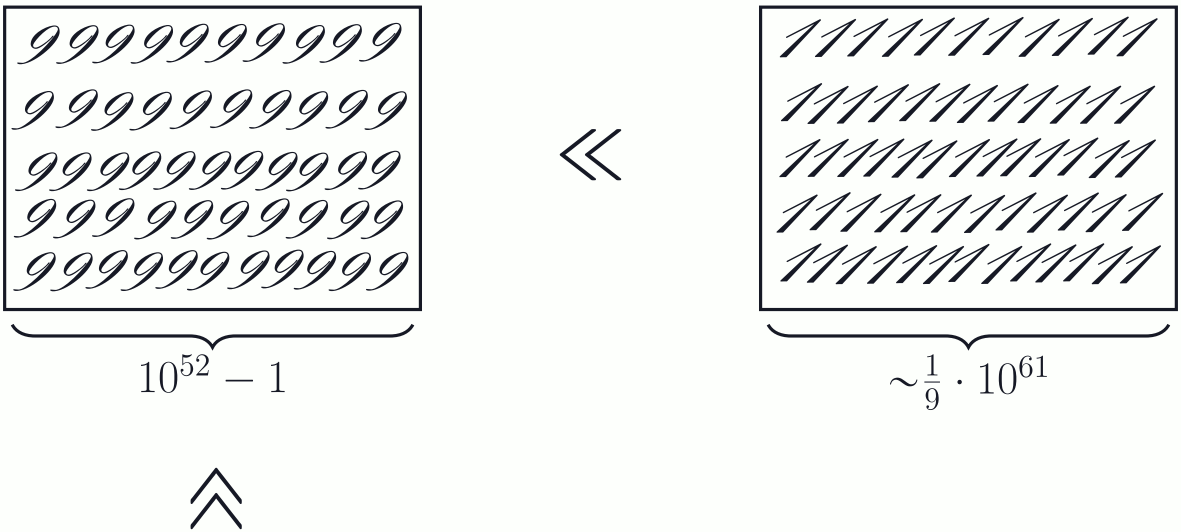
$\sim \frac{1}{9} \cdot 10^{61}$

$\Uparrow$

99999^99999

“**Who Can Name the Bigger Number?**” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html

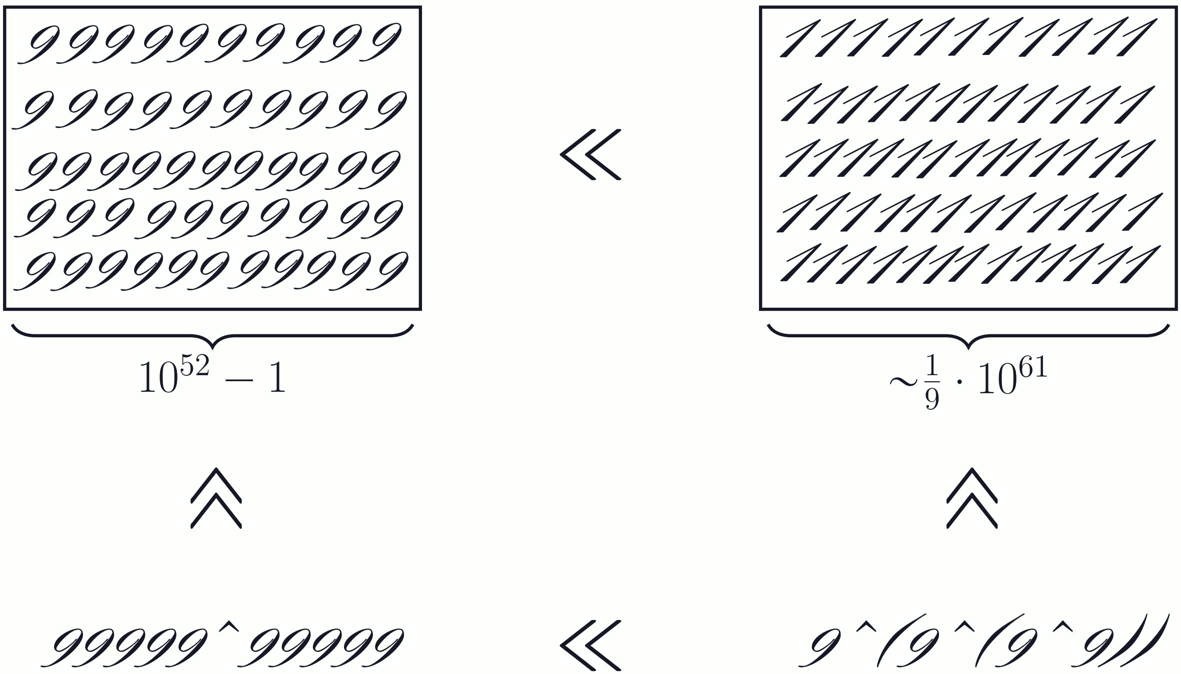


99999^99999

9^(9^(9^9))

“Who Can Name the Bigger Number?” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html



“**Who Can Name the Bigger Number?**” (Scott Aaronson)

https://www.scottaaronson.com/writings/bignumbers.html

$$\underbrace{\boxed{\begin{array}{l} 9999999999 \\ 9999999999 \\ 99999999999 \\ 9999999999 \\ 99999999999 \end{array}}}_{10^{52}-1} \ll \underbrace{\boxed{\begin{array}{l} 11111111111 \\ 111111111111 \\ 1111111111111 \\ 111111111111 \\ 1111111111111 \end{array}}}_{\sim\frac{1}{9}\cdot 10^{61}}$$

$$\Uparrow \qquad\qquad\qquad \Uparrow$$

$$\text{99999\^{}99999} \ll \text{9\^{}(9\^{}(9\^{}9))}$$

$\rightsquigarrow$ nie chodzi o same liczby tylko sposoby ich konstrukcji

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \quad := \underbrace{a \cdot \ldots \cdot a}_{b} \quad = a^b$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b := \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\}b$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \quad := \underbrace{a \cdot \ldots \cdot a}_{b} \quad = a^b$$

$$a \uparrow\uparrow b := \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\} b$$

$$\vdots$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \quad := \underbrace{a \cdot \ldots \cdot a}_{b} \quad = a^b$$

$$a \uparrow\uparrow b := \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} b \qquad a \uparrow^n b := \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \;:= \underbrace{a \cdot \ldots \cdot a}_{b} = a^b$$

$$a \uparrow\uparrow b := \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} b \qquad a \uparrow^n b := \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \quad := \underbrace{a \cdot \ldots \cdot a}_{b} \quad = a^b$$

$$a \uparrow\uparrow b := \underbrace{a \uparrow \ldots \uparrow a}_{b} = a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\Big\}\, b \qquad a \uparrow^n b := \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

$$\triangle(n) \quad := \quad n^n$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

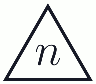
Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b \;:=\; \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} b \qquad a \uparrow^{n} b \;:=\; \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

 $:= \quad n^n$

$\boxed{n} \quad := \quad n$ w n trójkątach

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b \;:=\; \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\}b \qquad a \uparrow^n b \;:=\; \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

n $:=$ n^n

$\boxed{n}$ $:=$ n w n trójkątach

n $:=$ n w n kwadratach

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

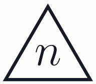
Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b \;:=\; \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} b \qquad a \uparrow^n b \;:=\; \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

 $:= \quad n^n$

$\boxed{n} \quad := \quad n$ w n trójkątach

 $:= \quad n$ w n kwadratach

$$\vdots$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \ := \underbrace{a \cdot \ldots \cdot a}_{b} \ = a^b$$

$$a \uparrow\uparrow b := \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\}b \qquad a \uparrow^n b := \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

$$n \quad := \quad n^n$$

$$\boxed{n} \quad := \quad n \text{ w } n \text{ trójkątach}$$

$$n \quad := \quad n \text{ w } n \text{ kwadratach}$$

$$\vdots$$

Liczby Grahama

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b \;:=\; \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} b \qquad a \uparrow^n b \;:=\; \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

$$\triangle(n) \;:=\; n^n$$

$$\boxed{n} \;:=\; n \text{ w } n \text{ trójkątach}$$

$$\pentagon(n) \;:=\; n \text{ w } n \text{ kwadratach}$$

$$\vdots$$

Liczby Grahama

$$G_1 \;:=\; 3 \uparrow^4 3$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b \;:=\; \underbrace{a \uparrow \ldots \uparrow a}_{b} = \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} b \qquad a \uparrow^n b \;:=\; \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

$$\triangle(n) \;:=\; n^n$$

$$\boxed{n} \;:=\; n \text{ w } n \text{ trójkątach}$$

$$\pentagon(n) \;:=\; n \text{ w } n \text{ kwadratach}$$

$$\vdots$$

Liczby Grahama

$$G_1 \;:=\; 3 \uparrow^4 3$$

$$G_{n+1} \;:=\; 3 \uparrow^{G_n} 3$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b \;:=\; \underbrace{a \uparrow \ldots \uparrow a}_{b} \;=\; \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\}b$$

$$\vdots$$

$$a \uparrow^n b \;:=\; \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} = \textbf{dużo}$$

Notacja Steinhausa–Mosera

$$\triangle n \;:=\; n^n$$

$$\boxed{n} \;:=\; n \text{ w } n \text{ trójkątach}$$

$$\pentagon n \;:=\; n \text{ w } n \text{ kwadratach}$$

$$\vdots$$

Liczby Grahama

$$G_1 \;:=\; 3 \uparrow^4 3$$

$$G_{n+1} \;:=\; 3 \uparrow^{G_n} 3$$

$$\vdots$$

Notacje

(czyli jak zapisać dużo operacji małą liczbą znaków)

Notacja strzałkowa Knutha

$$a \uparrow b \;:=\; \underbrace{a \cdot \ldots \cdot a}_{b} \;=\; a^b$$

$$a \uparrow\uparrow b \;:=\; \underbrace{a \uparrow \ldots \uparrow a}_{b} \;=\; \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} b$$

$$a \uparrow^n b \;:=\; \underbrace{a \uparrow^{n-1} \ldots \uparrow^{n-1} a}_{b} \;=\; \textbf{dużo}$$

$$\vdots$$

Notacja Steinhausa–Mosera

$$n \quad := \quad n^n$$

$$\boxed{n} \quad := \quad n \text{ w } n \text{ trójkątach}$$

$$n \quad := \quad n \text{ w } n \text{ kwadratach}$$

$$\vdots$$

Liczby Grahama

$$G_1 \quad := \; 3 \uparrow^4 3$$

$$G_{n+1} \quad := \; 3 \uparrow^{G_n} 3$$

$$\vdots$$

$$G_{64} \quad = \quad \textbf{bardzo dużo}$$

lePętlePętlePętlePętlePętlePętlePętlePętlePętlePętlePętlePętlePętlePęt

```
r := 1;
powtorz b razy {
    r := r * a;
}
```

$$
\left.\begin{array}{l}
\mathrm{r}\ :=\ 1; \\
\textbf{powtorz}\ \mathrm{b}\ \textbf{razy}\ \{ \\
\qquad \mathrm{r}\ :=\ \mathrm{r}\ *\ \mathrm{a}; \\
\}
\end{array}\right\}
\begin{array}{l}
r := a^b \\
= a \uparrow b
\end{array}
$$

```
b := 1;
powtorz c razy {
    r := 1;
    powtorz b razy {
        r := r * a;
    }
    b := r;
}
```

$r := a^b = a \uparrow b$

```
b := 1;
powtorz c razy {
    r := 1;
    powtorz b razy {
        r := r * a;
    }
    b := r;
}
```

$$\left.\begin{array}{l} r := a^b \\ = a \uparrow b \end{array}\right.$$

$$\left.\begin{array}{l} b := \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} c \\ = a \uparrow\uparrow c \end{array}\right.$$

```
b := 1;
powtorz c razy {
    r := 1;
    powtorz b razy {
        r := r * a;
    }
    b := r;
}
```

$r := a^b = a \uparrow b$

$b := \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} c = a \uparrow\uparrow c$

```
c := 1;
powtorz d razy {
    b := 1;
    powtorz c razy {
        r := 1;
        powtorz b razy {
            r := r * a;
        }
        b := r;
    }
    c := b;
}
```

$$r := a^b = a \uparrow b$$

$$b := \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\} c = a \uparrow\uparrow c$$

```
c := 1;
powtorz d razy {
    b := 1;
    powtorz c razy {
        r := 1;
        powtorz b razy {
            r := r * a;
        }
        b := r;
    }
    c := b;
}
```

$$r := a^b = a \uparrow b$$

$$b := \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\}c = a \uparrow\uparrow c$$

$$c := a \uparrow\uparrow\uparrow d$$

```
c := 1;
powtorz d razy {
    b := 1;
    powtorz c razy {
        r := 1;
        powtorz b razy {
            r := r * a;
        }
        b := r;
    }
    c := b;
}
```

$$r := a^b = a \uparrow b$$

$$b := \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\}c = a \uparrow\uparrow c$$

$$c := a \uparrow\uparrow\uparrow d$$

~~**dopoki** n > 0 **powtarzaj**~~

~~...~~

```
c := 1;
powtorz d razy {
    b := 1;
    powtorz c razy {
        r := 1;
        powtorz b razy {
            r := r * a;
        }
        b := r;
    }
    c := b;
}
```

$$r := a^b = a \uparrow b$$

$$b := \left.a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}\right\}c = a \uparrow\uparrow c$$

$$c := a \uparrow\uparrow\uparrow d$$

~~**dopoki** n > 0 **powtarzaj** . . .~~

$\rightsquigarrow$ funkcje **pierwotnie rekurencyjne**

```
c := 1;
powtorz d razy {
    b := 1;
    powtorz c razy {
        r := 1;
        powtorz b razy {
            r := r * a;
        }
        b := r;
    }
    c := b;
}
```

$r := a^b = a \uparrow b$

$b := \left. a^{a^{\cdot^{\cdot^{\cdot^{a}}}}} \right\} c = a \uparrow\uparrow c$

$c := a \uparrow\uparrow\uparrow d$

~~**dopoki** n > 0 **powtarzaj**~~

~~...~~

$\rightsquigarrow$ funkcje **pierwotnie rekurencyjne**

[nigdy się nie pętlą $\Rightarrow$ zawsze zwracają wynik]

Funkcja Ackermanna i Pétera i Robinsona

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0, n) := n + 1$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0, n) := n + 1$$

$$\mathbf{A}(m, 0) := \mathbf{A}\big(m - 1, 1\big)$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\langle m',n'\rangle <_{\mathrm{lex}} \langle m,n\rangle$.

Funkcja Ackermanna i Pétera i Robinsona

$$\begin{aligned} \mathbf{A}(0,n) &:= n+1 \\ \mathbf{A}(m,0) &:= \mathbf{A}\big(m-1,1\big) \\ \mathbf{A}(m,n) &:= \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big) \end{aligned}$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\mathrm{lex}} \langle m,n\rangle}_{\big(m'=m \wedge n'<n\big) \vee m'<m}$.

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\text{lex}} \langle m,n\rangle}_{(m'=m \,\wedge\, n'<n) \,\vee\, m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\text{lex}}\rangle$ jest **dobrze ufundowany**.

Funkcja Ackermanna i Pétera i Robinsona

$$\begin{aligned}\mathbf{A}(0,n) &:= n+1\\ \mathbf{A}(m,0) &:= \mathbf{A}\big(m-1,1\big)\\ \mathbf{A}(m,n) &:= \mathbf{A}\big(m-1,\mathbf{A}(m,n{-}1)\big)\end{aligned}$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\text{lex}} \langle m,n\rangle}_{\big(m'=m \,\wedge\, n'<n\big) \,\vee\, m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\text{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ∎

Funkcja Ackermanna i Pétera i Robinsona

$$\begin{aligned} \mathbf{A}(0,n) &:= n+1 \\ \mathbf{A}(m,0) &:= \mathbf{A}\big(m-1,1\big) \\ \mathbf{A}(m,n) &:= \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big) \end{aligned}$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\text{lex}} \langle m,n\rangle}_{(m'=m \wedge n'<n) \vee m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\text{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\mathbf{A}(0,n) = n+1$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\mathrm{lex}} \langle m,n\rangle}_{(m'=m \wedge n'<n) \vee m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\mathrm{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\mathbf{A}(0,n) = n+1$$
$$\mathbf{A}(1,n) = n+2$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\text{lex}} \langle m,n\rangle}_{(m'=m \wedge n'<n) \vee m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\text{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\mathbf{A}(0,n) = n+1$$
$$\mathbf{A}(1,n) = n+2$$
$$\mathbf{A}(2,n) = 2n+3$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\text{lex}} \langle m,n\rangle}_{\big(m'=m \,\wedge\, n'<n\big) \,\vee\, m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\text{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\mathbf{A}(0,n) = n+1$$
$$\mathbf{A}(1,n) = n+2$$
$$\mathbf{A}(2,n) = 2n+3$$
$$\mathbf{A}(3,n) = 2^{n+3}-3$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\mathrm{lex}} \langle m,n\rangle}_{(m'=m \,\wedge\, n'<n) \,\vee\, m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\mathrm{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\mathbf{A}(0,n) = n+1$$
$$\mathbf{A}(1,n) = n+2$$
$$\mathbf{A}(2,n) = 2n+3$$
$$\mathbf{A}(3,n) = 2^{n+3}-3$$
$$\vdots$$

Funkcja Ackermanna i Pétera i Robinsona

$$\begin{aligned}\mathbf{A}(0,n) &:= n+1\\ \mathbf{A}(m,0) &:= \mathbf{A}\big(m-1,1\big)\\ \mathbf{A}(m,n) &:= \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)\end{aligned}$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\text{lex}} \langle m,n\rangle}_{(m'=m \,\wedge\, n'<n) \,\vee\, m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\text{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\begin{aligned}\mathbf{A}(0,n) &= n+1\\ \mathbf{A}(1,n) &= n+2\\ \mathbf{A}(2,n) &= 2n+3\\ \mathbf{A}(3,n) &= 2^{n+3}-3\\ &\vdots\\ \mathbf{A}(m,n) &\sim 2\uparrow^m n\end{aligned}$$

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\text{lex}} \langle m,n\rangle}_{(m'=m \,\wedge\, n'<n) \,\vee\, m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\text{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\mathbf{A}(0,n) = n+1$$
$$\mathbf{A}(1,n) = n+2$$
$$\mathbf{A}(2,n) = 2n+3$$
$$\mathbf{A}(3,n) = 2^{n+3}-3$$
$$\vdots$$
$$\mathbf{A}(m,n) \sim 2\uparrow^{m} n$$

Twierdzenie (Ackermann [1928])

Funkcja **Ackermanna nie jest pierwotnie rekurencyjna**.

Funkcja Ackermanna i Pétera i Robinsona

$$\mathbf{A}(0,n) := n+1$$
$$\mathbf{A}(m,0) := \mathbf{A}\big(m-1,1\big)$$
$$\mathbf{A}(m,n) := \mathbf{A}\big(m-1,\mathbf{A}(m,n-1)\big)$$

Czy $\mathbf{A}(m,n)$ jest dobrze określona?

- $\mathbf{A}(m,n)$ zależy od $\mathbf{A}(m',n')$ dla $\underbrace{\langle m',n'\rangle <_{\mathrm{lex}} \langle m,n\rangle}_{\big(m'=m \,\wedge\, n'<n\big) \,\vee\, m'<m}$.
- porządek $\langle \mathbb{N}\times\mathbb{N}, \leqslant_{\mathrm{lex}}\rangle$ jest **dobrze ufundowany**.
- **indukcja**: „$\mathbf{A}(m,n)$ jest dobrze określone”. ■

$$\mathbf{A}(0,n) = n+1$$
$$\mathbf{A}(1,n) = n+2$$
$$\mathbf{A}(2,n) = 2n+3$$
$$\mathbf{A}(3,n) = 2^{n+3}-3$$
$$\vdots$$
$$\mathbf{A}(m,n) \sim 2\uparrow^{m} n$$

Twierdzenie (Ackermann [1928])

Funkcja **Ackermanna nie jest pierwotnie rekurencyjna**.

Dowód

Indukcja po zagnieżdżeniu pętli m:

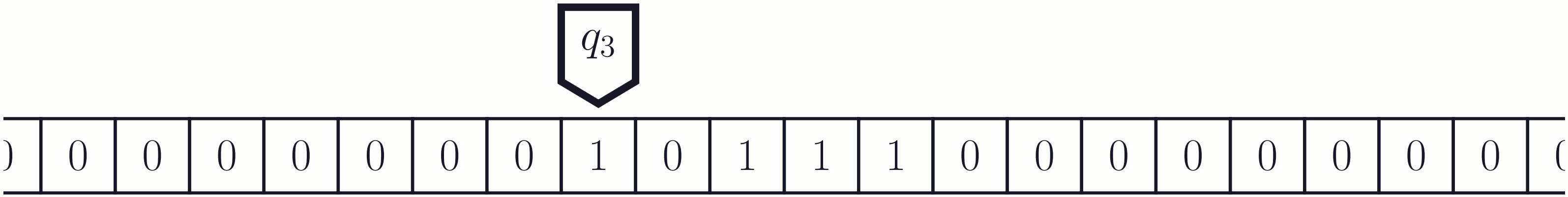
$\mathbf{A}(m,\_)$ rośnie szybciej niż $f(\_)$. ■

Maszyny Turinga

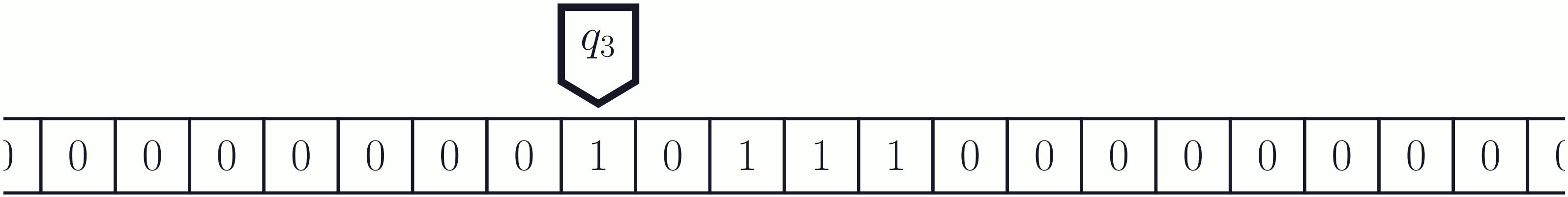
Maszyny Turinga

0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0

Maszyny Turinga



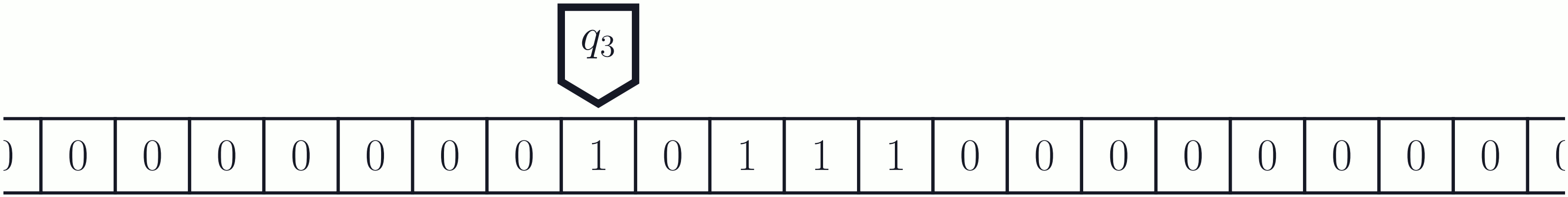
Maszyny Turinga



$\mathcal{M}$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

Maszyny Turinga

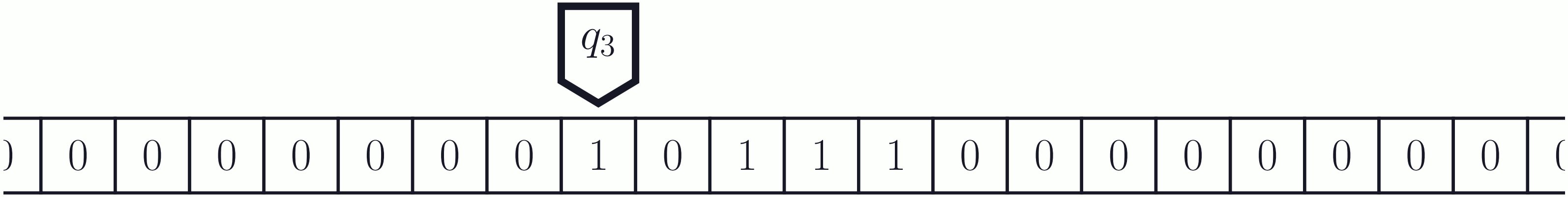


$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Maszyny Turinga

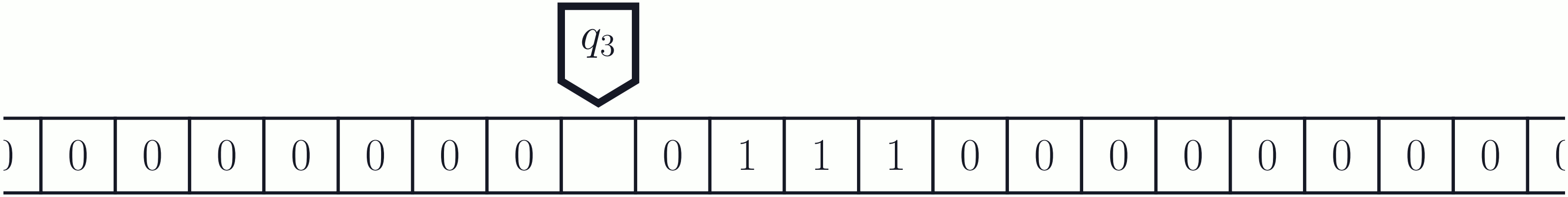


$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Maszyny Turinga

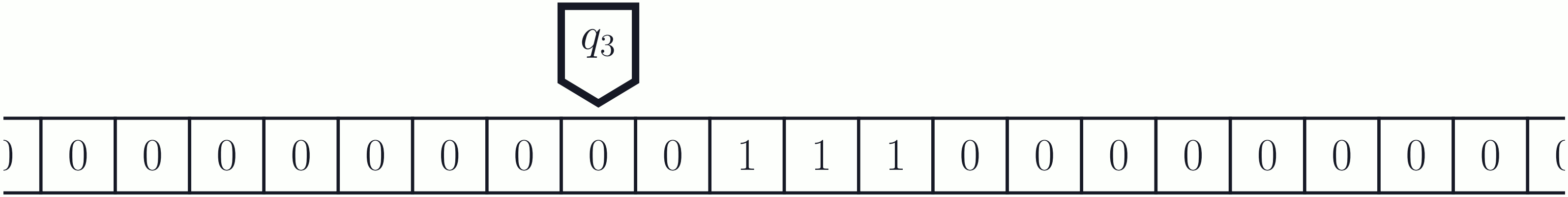


$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \longmapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \longmapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \longmapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \longmapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \longmapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \longmapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Maszyny Turinga

q_5

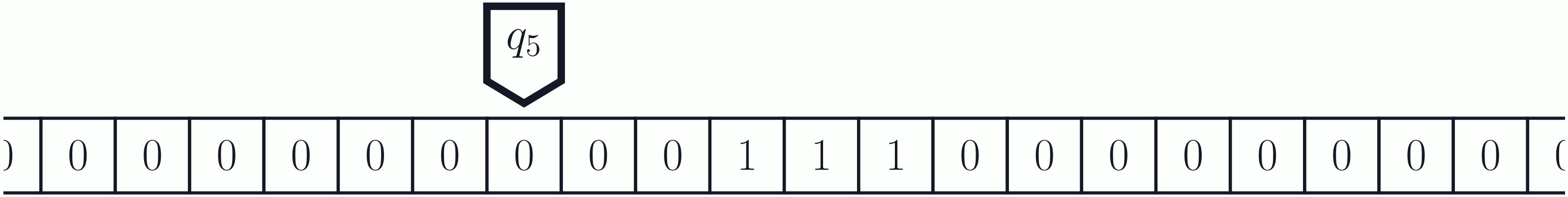
0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Maszyny Turinga



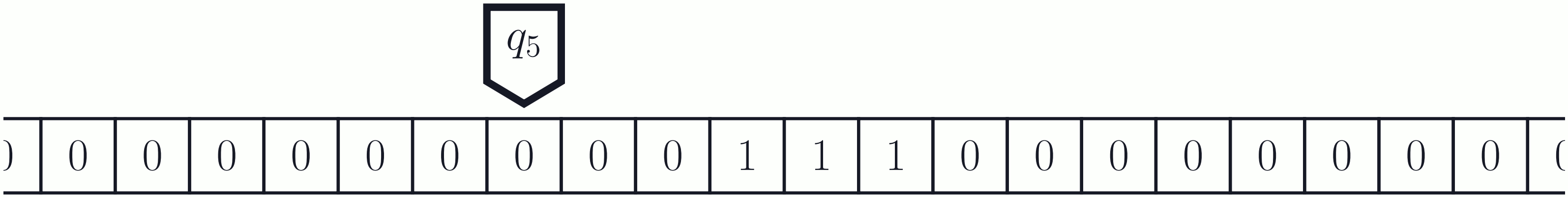
$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $|\mathcal{M}| := n$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

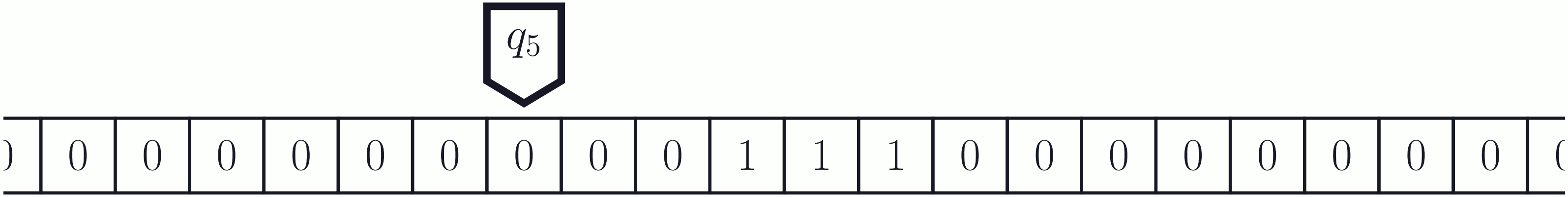
$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $\big|\mathcal{M}\big| := n$

Czas: $\mathrm{time}\big(\mathcal{M}\big) \in \mathbb{N} \cup \{\infty\}$

$\big[$ początkowo $\cdots 0000 \cdots$ $\big]$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $\big|\mathcal{M}\big| := n$

Czas: $\mathrm{time}\big(\mathcal{M}\big) \in \mathbb{N} \cup \{\infty\}$

$\big[$ początkowo $\cdots 0000 \cdots$ $\big]$

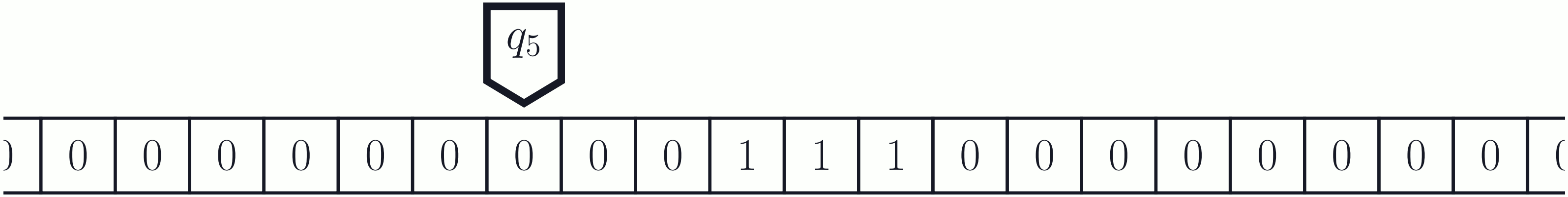
Fakt

Dla każdego $n \in \mathbb{N}$

jest tylko **skończenie wiele**

maszyn rozmiaru n.

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \longmapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \longmapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \longmapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $\left|\mathcal{M}\right| := n$

Czas: $\mathrm{time}\big(\mathcal{M}\big) \in \mathbb{N} \cup \{\infty\}$

$\big[$ początkowo $\cdots 0000 \cdots$ $\big]$

Fakt

Dla każdego $n \in \mathbb{N}$

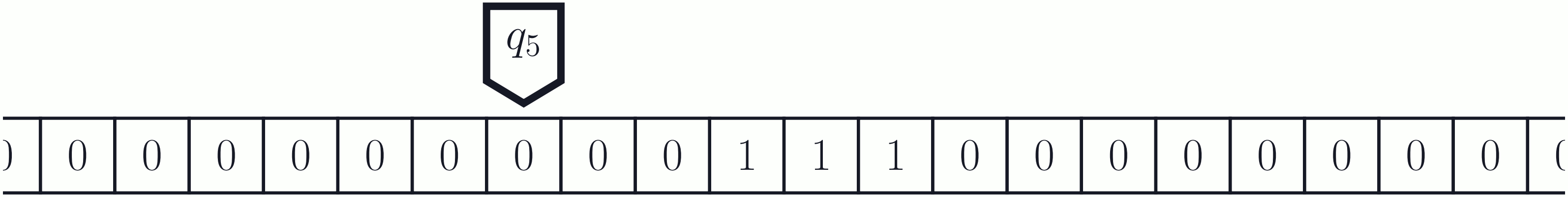
jest tylko **skończenie wiele**

maszyn rozmiaru n.

Busy Beaver (pracowity bóbr):

$$\mathbf{BB}(n) := \max \Big\{ \mathrm{time}(\mathcal{M}) \,\Big|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \Big\} \in \mathbb{N}$$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $\big|\mathcal{M}\big| := n$

Czas: $\mathrm{time}\big(\mathcal{M}\big) \in \mathbb{N} \cup \{\infty\}$

$\big[$ początkowo $\cdots 0000 \cdots$ $\big]$

BB$(1) = \_$

Fakt

Dla każdego $n \in \mathbb{N}$

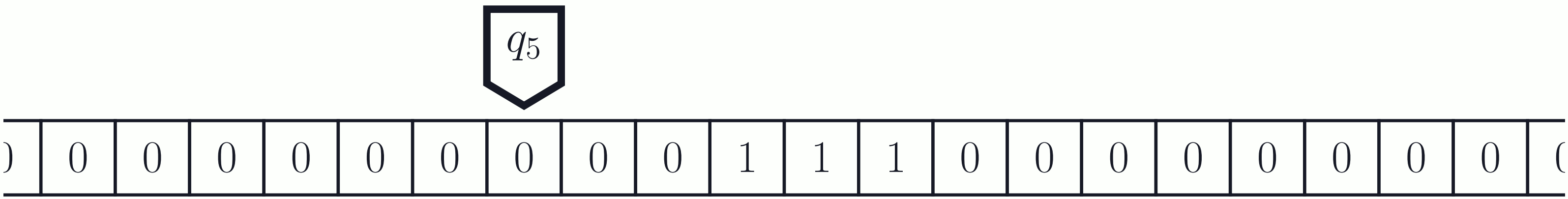
jest tylko **skończenie wiele**

maszyn rozmiaru n.

Busy Beaver (pracowity bóbr):

$$\mathbf{BB}(n) := \max\left\{ \mathrm{time}(\mathcal{M}) \;\middle|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \right\} \in \mathbb{N}$$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $\big|\mathcal{M}\big| := n$

Czas: $\mathrm{time}\big(\mathcal{M}\big) \in \mathbb{N} \cup \{\infty\}$

$\Big[$ początkowo $\cdots 0000 \cdots$ $\Big]$

$\mathbf{BB}(1) = \_$

$\mathbf{BB}(2) = 6$

Fakt

Dla każdego $n \in \mathbb{N}$

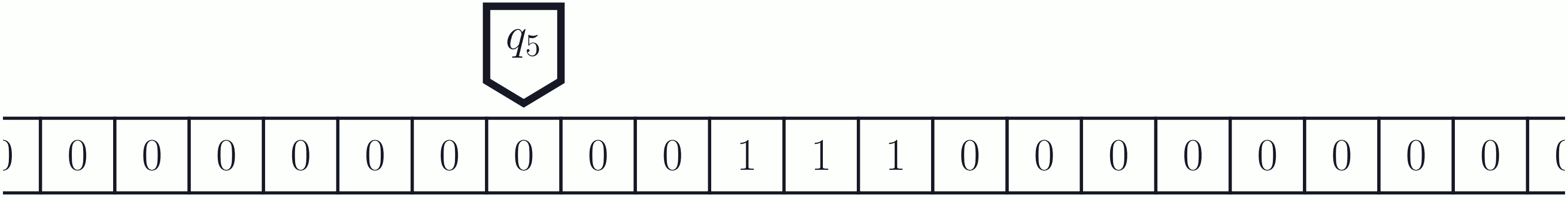
jest tylko **skończenie wiele**

maszyn rozmiaru n.

Busy Beaver (pracowity bóbr):

$$\mathbf{BB}(n) := \max \Big\{ \mathrm{time}(\mathcal{M}) \,\Big|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \Big\} \in \mathbb{N}$$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $|\mathcal{M}| := n$

Czas: $\mathrm{time}(\mathcal{M}) \in \mathbb{N} \cup \{\infty\}$

$\left[\ \text{początkowo}\ \cdots 0000 \cdots\ \right]$

Fakt

Dla każdego $n \in \mathbb{N}$ jest tylko **skończenie wiele** maszyn rozmiaru n.

BB$(1) = \_$

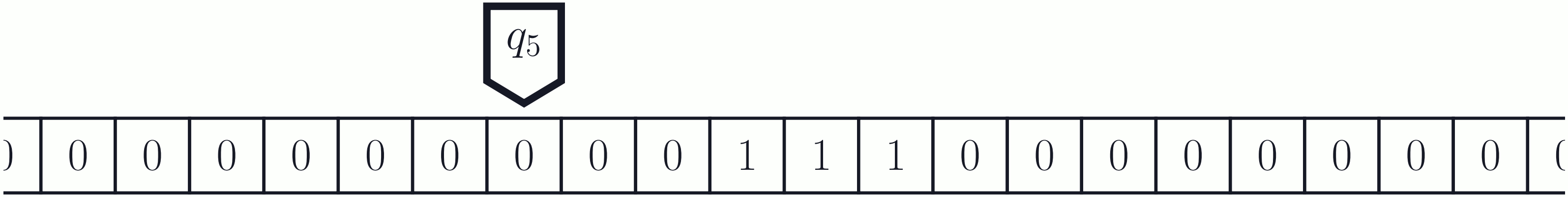
BB$(2) = 6$

BB$(3) = 21$

Busy Beaver (pracowity bóbr):

$$\mathbf{BB}(n) := \max\left\{ \mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \right\} \in \mathbb{N}$$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $|\mathcal{M}| := n$

Czas: $\mathrm{time}(\mathcal{M}) \in \mathbb{N} \cup \{\infty\}$

[początkowo $\cdots 0000 \cdots$]

Fakt

Dla każdego $n \in \mathbb{N}$

jest tylko **skończenie wiele**

maszyn rozmiaru n.

$\mathbf{BB}(1) = \_$

$\mathbf{BB}(2) = 6$

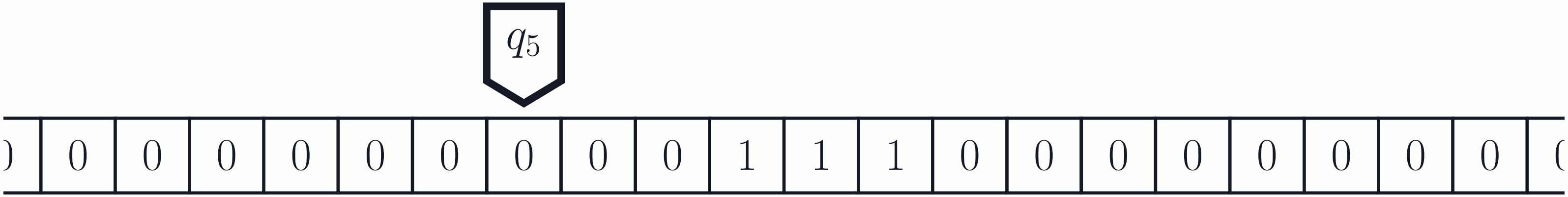
$\mathbf{BB}(3) = 21$

$\mathbf{BB}(4) = 107$

Busy Beaver (pracowity bóbr):

$$\mathbf{BB}(n) := \max \left\{ \mathrm{time}(\mathcal{M}) \;\middle|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \right\} \in \mathbb{N}$$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $\left|\mathcal{M}\right| := n$

Czas: $\mathrm{time}\big(\mathcal{M}\big) \in \mathbb{N} \cup \{\infty\}$

$\big[$ początkowo $\cdots 0000 \cdots$ $\big]$

Fakt

Dla każdego $n \in \mathbb{N}$

jest tylko **skończenie wiele**

maszyn rozmiaru n.

BB$(1) = \_$

BB$(2) = 6$

BB$(3) = 21$

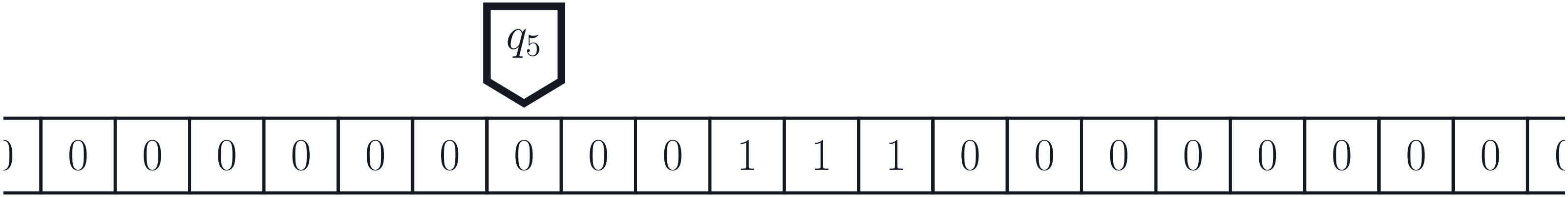
BB$(4) = 107$

BB$(5) = ?$

Busy Beaver (pracowity bóbr):

$$\mathbf{BB}(n) := \max \Big\{ \mathrm{time}(\mathcal{M}) \,\Big|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \Big\} \in \mathbb{N}$$

Maszyny Turinga



$$\mathcal{M}$$

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

$$\Delta = \left\{ \begin{array}{c} \vdots \\ q_3, 0 \mapsto 0, \blacktriangleright, q_4 \\ q_3, 1 \mapsto 0, \blacktriangleleft, q_5 \\ q_4, 0 \mapsto 0, \blacktriangleleft, q_4 \\ \vdots \end{array} \right\}$$

Rozmiar: $\big|\mathcal{M}\big| := n$

Czas: $\mathrm{time}\big(\mathcal{M}\big) \in \mathbb{N} \cup \{\infty\}$

$\Big[$ początkowo $\cdots 0000 \cdots$ $\Big]$

Fakt

Dla każdego $n \in \mathbb{N}$

jest tylko **skończenie wiele**

maszyn rozmiaru n.

$\mathbf{BB}(1) = \_$

$\mathbf{BB}(2) = 6$

$\mathbf{BB}(3) = 21$

$\mathbf{BB}(4) = 107$

$\mathbf{BB}(5) = ?$

$(\geqslant 47'176'870)$

Busy Beaver (pracowity bóbr):

$$\mathbf{BB}(n) := \max \Big\{ \mathrm{time}(\mathcal{M}) \;\Big|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \Big\} \in \mathbb{N}$$

$\mathrm{BB}(n)$ rośnie szybko

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

- wypisuje na pustej taśmie liczbę n

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

- wypisuje na pustej taśmie liczbę n
- uruchamia $\mathcal{M}$ by obliczyć $k := f(n)$

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

- wypisuje na pustej taśmie liczbę n
- uruchamia $\mathcal{M}$ by obliczyć $k := f(n)$
- czeka przynajmniej k kroków (modyfikując taśmę)

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

- wypisuje na pustej taśmie liczbę n
- uruchamia $\mathcal{M}$ by obliczyć $k := f(n)$
- czeka przynajmniej k kroków (modyfikując taśmę)
- kończy pracę

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

- wypisuje na pustej taśmie liczbę n
- uruchamia $\mathcal{M}$ by obliczyć $k := f(n)$
- czeka przynajmniej k kroków (modyfikując taśmę)
- kończy pracę

3. $|\mathcal{M}'_n| \leqslant \log n + C + |\mathcal{M}|$

$\mathbf{BB}(n)$ **rośnie szybko**

Twierdzenie

Jeśli $f : \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

- wypisuje na pustej taśmie liczbę n
- uruchamia $\mathcal{M}$ by obliczyć $k := f(n)$
- czeka przynajmniej k kroków (modyfikując taśmę)
- kończy pracę

3. $\left|\mathcal{M}'_n\right| \leqslant \log n + C + \left|\mathcal{M}\right| \leqslant n$ (d.d.d. n)

$\mathbf{BB}(n)$ rośnie szybko

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która:

- wypisuje na pustej taśmie liczbę n
- uruchamia $\mathcal{M}$ by obliczyć $k := f(n)$
- czeka przynajmniej k kroków (modyfikując taśmę)
- kończy pracę

3. $|\mathcal{M}'_n| \leqslant \log n + C + |\mathcal{M}| \leqslant n$ (d.d.d. n)

4. $\mathbf{BB}(n) \geqslant \mathrm{time}(\mathcal{M}'_n) > f(n)$ (d.d.d. n)

$\mathbf{BB}(n)$ **rośnie szybko**

Twierdzenie

Jeśli $f\colon \mathbb{N} \to \mathbb{N}$ jest obliczalna to d.d.d. n mamy $\mathbf{BB}(n) \gg f(n)$.

Dowód

1. Niech $\mathcal{M}$ będzie **maszyną Turinga** obliczającą f:

$$\forall n \in \mathbb{N}.\ \mathcal{M}[n] = f(n)$$

2. Niech $\mathcal{M}'_n$ będzie **maszyną Turinga** która: [złożoność Kołmogorowa]

- wypisuje na pustej taśmie liczbę n
- uruchamia $\mathcal{M}$ by obliczyć $k := f(n)$
- czeka przynajmniej k kroków (modyfikując taśmę)
- kończy pracę

3. $\left|\mathcal{M}'_n\right| \leqslant \log n + C + \left|\mathcal{M}\right| \leqslant n$ (d.d.d. n)

4. $\mathbf{BB}(n) \geqslant \mathrm{time}(\mathcal{M}'_n) > f(n)$ (d.d.d. n)

Problem stopu

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$? $\equiv \left[\text{czy } \mathcal{M} \text{ uruchomiona na } \cdots 000 \cdots \text{ się zatrzyma?} \right]$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$? $\equiv \Big[$ czy $\mathcal{M}$ uruchomiona na $\cdots 000 \cdots$ się zatrzyma? $\Big]$

Twierdzenie (Turing [1936])

Problem stopu jest nieobliczalny.

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$? $\equiv \Big[$ czy $\mathcal{M}$ uruchomiona na $\cdots 000 \cdots$ się zatrzyma? $\Big]$

Twierdzenie (Turing [1936])

Problem stopu jest nieobliczalny.

Dowód

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\text{time}(\mathcal{M}) < \infty$? $\equiv \Big[$ czy $\mathcal{M}$ uruchomiona na $\cdots 000 \cdots$ się zatrzyma? $\Big]$

Twierdzenie (Turing [1936])

Problem stopu jest nieobliczalny.

Dowód

Kodowanie paradoksu kłamcy:

$\mathcal{M}$ która się zatrzymuje wtw. gdy $\mathcal{M}$ się **nie** zatrzymuje

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$? $\equiv \left[\text{czy } \mathcal{M} \text{ uruchomiona na } \cdots 000 \cdots \text{ się zatrzyma?} \right]$

Twierdzenie (Turing [1936])

Problem stopu jest nieobliczalny.

Dowód

Kodowanie paradoksu kłamcy:

$\mathcal{M}$ która się zatrzymuje wtw. gdy $\mathcal{M}$ się **nie** zatrzymuje

$\rightsquigarrow$ Maszyna może się **pętlić** na niejeden sposób. . .

$\mathrm{BB}(n)$ jest <u>nieobliczalna</u>

$\mathbf{BB}(n)$ jest nieobliczalna

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \;\middle|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

$\mathbf{BB}(n)$ jest nieobliczalna

$$\mathbf{BB}(n) := \max \left\{ \mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

BB(n) jest nieobliczalna

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja **BB**(n) jest nieobliczalna.

Dowód

$\mathbf{BB}(n)$ jest nieobliczalna

$$\mathbf{BB}(n) := \max \left\{ \mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Dowód

Załóżmy, że umiemy policzyć $\mathbf{BB}(n)$. Rozwiążmy **problem stopu**:

$\mathbf{BB}(n)$ jest nieobliczalna

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \;\middle|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Dowód

Załóżmy, że umiemy policzyć $\mathbf{BB}(n)$. Rozwiążmy **problem stopu**:

- wczytajmy opis maszyny $\mathcal{M}$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{ \mathrm{time}(\mathcal{M}) \;\middle|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty \right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Dowód

Załóżmy, że umiemy policzyć $\mathbf{BB}(n)$. Rozwiążmy **problem stopu**:

- wczytajmy opis maszyny $\mathcal{M}$
- obliczmy $n := \left|\mathcal{M}\right|$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Dowód

Załóżmy, że umiemy policzyć $\mathbf{BB}(n)$. Rozwiążmy **problem stopu**:

- wczytajmy opis maszyny $\mathcal{M}$
- obliczmy $n := \left|\mathcal{M}\right|$
- obliczmy $T := \mathbf{BB}(n)$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \;\middle|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Dowód

Załóżmy, że umiemy policzyć $\mathbf{BB}(n)$. Rozwiążmy **problem stopu**:

- wczytajmy opis maszyny $\mathcal{M}$
- obliczmy $n := |\mathcal{M}|$
- obliczmy $T := \mathbf{BB}(n)$
- uruchommy $\mathcal{M}$ na $T+1$ kroków:

$\mathbf{BB}(n)$ jest nieobliczalna

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Dowód

Załóżmy, że umiemy policzyć $\mathbf{BB}(n)$. Rozwiążmy **problem stopu**:

- wczytajmy opis maszyny $\mathcal{M}$
- obliczmy $n := |\mathcal{M}|$
- obliczmy $T := \mathbf{BB}(n)$
- uruchommy $\mathcal{M}$ na $T+1$ kroków:

  a. $\mathcal{M}$ skończyła pracę $\implies \mathrm{time}(\mathcal{M}) < \infty$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \;\middle|\; |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Dowód

Załóżmy, że umiemy policzyć $\mathbf{BB}(n)$. Rozwiążmy **problem stopu**:

- wczytajmy opis maszyny $\mathcal{M}$
- obliczmy $n := |\mathcal{M}|$
- obliczmy $T := \mathbf{BB}(n)$
- uruchommy $\mathcal{M}$ na $T+1$ kroków:

  a. $\mathcal{M}$ skończyła pracę $\implies \mathrm{time}(\mathcal{M}) < \infty$

  b. $\mathcal{M}$ nie skończyła pracy $\implies \mathrm{time}(\mathcal{M}) > \mathbf{BB}(n) \implies \mathrm{time}(\mathcal{M}) = \infty$

$\mathbf{BB}(n)$ jest nieobliczalna

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

Dane: opis maszyny Turinga $\mathcal{M}$

Wynik: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

BB(n) jest nieobliczalna

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja **BB**(n) jest nieobliczalna.

Twierdzenie (Pavlus [2020])

Istnieje $\mathcal{M}$ o 748 stanach, taka że:

$$\mathrm{time}(\mathcal{M}) < \infty \iff \text{teoria } \mathbf{ZFC} \text{ jest sprzeczna}$$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Twierdzenie (Pavlus [2020])

Istnieje $\mathcal{M}$ o 748 stanach, taka że:

$$\mathrm{time}(\mathcal{M}) < \infty \iff \text{teoria } \mathbf{ZFC} \text{ jest sprzeczna}$$

Twierdzenie (Aaronson [2016]; Pavlus [2020])

Istnieje $\mathcal{M}$ o 744 stanach, taka że:

$$\mathrm{time}(\mathcal{M}) < \infty \iff \text{hipoteza Riemanna jest fałszywa}$$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{\text{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \text{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\text{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Twierdzenie (Pavlus [2020])

Istnieje $\mathcal{M}$ o 748 stanach, taka że:

$$\text{time}(\mathcal{M}) < \infty \iff \text{teoria } \mathbf{ZFC} \text{ jest sprzeczna}$$

Twierdzenie (Aaronson [2016]; Pavlus [2020])

Istnieje $\mathcal{M}$ o 744 stanach, taka że:

$$\text{time}(\mathcal{M}) < \infty \iff \text{hipoteza Riemanna jest fałszywa}$$

$\vdots$

$\mathbf{BB}(n)$ jest <u>nieobliczalna</u>

$$\mathbf{BB}(n) := \max\left\{\mathrm{time}(\mathcal{M}) \,\middle|\, |\mathcal{M}| \leqslant n, \mathrm{time}(\mathcal{M}) < \infty\right\} \in \mathbb{N}$$

Problem stopu

DANE: opis maszyny Turinga $\mathcal{M}$

WYNIK: czy $\mathrm{time}(\mathcal{M}) < \infty$?

Twierdzenie (Radó [1962])

Funkcja $\mathbf{BB}(n)$ jest nieobliczalna.

Twierdzenie (Pavlus [2020])

Istnieje $\mathcal{M}$ o 748 stanach, taka że:

$$\mathrm{time}(\mathcal{M}) < \infty \iff \text{teoria } \mathbf{ZFC} \text{ jest sprzeczna}$$

Twierdzenie (Aaronson [2016]; Pavlus [2020])

Istnieje $\mathcal{M}$ o 744 stanach, taka że:

$$\mathrm{time}(\mathcal{M}) < \infty \iff \text{hipoteza Riemanna jest fałszywa}$$

$\vdots$

+ kamienne tablice

Paradoks kłamcy

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować
w języku polskim z użyciem 100 symboli"

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować
w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować
w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \;\middle|\; \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \varphi_k(i) \iff i = n \right\}$$

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować
w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \;\middle|\; \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \varphi_k(i) \iff i = n \right\} + 1$$

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować
w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max\left\{ n \in \mathbb{N} \;\middle|\; \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \Longleftrightarrow i = n \right\} + 1$$

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować
w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \;\middle|\; \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n \right\} + 1$$

$\mathtt{TRUTH}(k, i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \mid \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \land \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n \right\} + 1$$

$\mathtt{TRUTH}(k, i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Twierdzenie (Tarski [1933]) *(o niedefiniowalności prawdy)*

Własność $\mathtt{TRUTH}(k, i)$ jest **niedefiniowalna** żadną formułą logiki.

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \mid \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n \right\} + 1$$

$\mathtt{TRUTH}(k, i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Twierdzenie (Tarski [1933]) *(o niedefiniowalności prawdy)*

Własność $\mathtt{TRUTH}(k, i)$ jest **niedefiniowalna** żadną formułą logiki.

Intuicja

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \mid \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \land \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n \right\} + 1$$

$\mathtt{TRUTH}(k,i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Twierdzenie (Tarski [1933]) (*o niedefiniowalności prawdy*)

Własność $\mathtt{TRUTH}(k,i)$ jest **niedefiniowalna** żadną formułą logiki.

Intuicja

Załóżmy, że $\mathtt{TRUTH}(k,i)$ jest definiowane formułą $\tau(k,i)$.

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \mid \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n \right\} + 1$$

$\mathtt{TRUTH}(k, i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Twierdzenie (Tarski [1933]) *(o niedefiniowalności prawdy)*

Własność $\mathtt{TRUTH}(k, i)$ jest **niedefiniowalna** żadną formułą logiki.

Intuicja

Załóżmy, że $\mathtt{TRUTH}(k, i)$ jest definiowane formułą $\tau(k, i)$.
Każda formuła $\varphi_k(i)$ jest równoważna $\tau(k, i)$.

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \Big\{ n \in \mathbb{N} \;\Big|\; \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n \Big\} + 1$$

$\mathtt{TRUTH}(k, i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Twierdzenie (Tarski [1933]) *(o niedefiniowalności prawdy)*

Własność $\mathtt{TRUTH}(k, i)$ jest **niedefiniowalna** żadną formułą logiki.

Intuicja

Załóżmy, że $\mathtt{TRUTH}(k, i)$ jest definiowane formułą $\tau(k, i)$.
Każda formuła $\varphi_k(i)$ jest równoważna $\tau(k, i)$.
Formuła τ ma określoną liczbę kwantyfikatorów, np. 7.

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max\left\{n \in \mathbb{N} \mid \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \land \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n\right\} + 1$$

$\mathtt{TRUTH}(k,i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Twierdzenie (Tarski [1933]) (*o niedefiniowalności prawdy*)

Własność $\mathtt{TRUTH}(k,i)$ jest **niedefiniowalna** żadną formułą logiki.

Intuicja

Załóżmy, że $\mathtt{TRUTH}(k,i)$ jest definiowane formułą $\tau(k,i)$.
Każda formuła $\varphi_k(i)$ jest równoważna $\tau(k,i)$.
Formuła τ ma określoną liczbę kwantyfikatorów, np. 7.
Więc każdą własność da się wyrazić z użyciem 7 kwantyfikatorów.

Paradoks kłamcy

$n :=$ „największa liczba naturalna, którą można zdefiniować w języku polskim z użyciem 100 symboli; dodać 1”

Ale przecież daje się enumerować formuły arytmetyki: $\varphi_0, \varphi_1, \ldots$

$$n := \max \left\{ n \in \mathbb{N} \mid \exists k \in \mathbb{N}.\ |\varphi_k| < 100 \wedge \forall i \in \mathbb{N}.\ \underbrace{\varphi_k(i)}_{\mathtt{TRUTH}(k,i)} \iff i = n \right\} + 1$$

$\mathtt{TRUTH}(k, i) \equiv$ „formuła o numerze k jest prawdziwa dla argumentu i”

Twierdzenie (Tarski [1933]) *(o niedefiniowalności prawdy)*

Własność $\mathtt{TRUTH}(k, i)$ jest **niedefiniowalna** żadną formułą logiki.

Intuicja

Załóżmy, że $\mathtt{TRUTH}(k, i)$ jest definiowane formułą $\tau(k, i)$.
Każda formuła $\varphi_k(i)$ jest równoważna $\tau(k, i)$.
Formuła τ ma określoną liczbę kwantyfikatorów, np. 7.
Więc każdą własność da się wyrazić z użyciem 7 kwantyfikatorów.
Sprzeczność!

Zajęcia praktyczno-techniczne

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z`

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, . . .

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, . . .

Polecenia:

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, ..., `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, ...

Polecenia:

- `inc x;` $\equiv$ `x := x + 1;`

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, . . .

Polecenia:

- `inc x;` $\equiv$ `x := x + 1;`
- `dec x;` $\equiv$ `x := x - 1;`

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, . . .

Polecenia:

- `inc x;` ≡ `x := x + 1;`
- `dec x;` ≡ `x := x - 1;`
- `jnz x` i`;` ≡ `if x <> 0 then goto` i`;`

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, . . .

Polecenia:

- `inc x;` $\equiv$ `x := x + 1;`
- `dec x;` $\equiv$ `x := x - 1;`
- `jnz x` i`;` $\equiv$ `if x <> 0 then goto` i`;`

Program: ciąg maksymalnie 21 instrukcji

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, ..., `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, ...

Polecenia:

- `inc x;` $\equiv$ `x := x + 1;`
- `dec x;` $\equiv$ `x := x - 1;`
- `jnz x` i`;` $\equiv$ `if x <> 0 then goto` i`;`

Program: ciąg maksymalnie 21 instrukcji

0: **inc** a ;
1: **inc** a ;
2: **inc** a ;
3: **dec** a ;
4: **inc** b ;
5: **inc** b ;
6: **inc** b ;
7: **jnz** a 3 ;

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, ..., `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, ...

Polecenia:

- `inc x;` $\equiv$ `x := x + 1;`
- `dec x;` $\equiv$ `x := x - 1;`
- `jnz x` i`;` $\equiv$ `if x <> 0 then goto` i`;`

Program: ciąg maksymalnie 21 instrukcji

Wynik: maksymalna wartość zmiennej po **zakończeniu** działania

```
0: inc a;
1: inc a;
2: inc a;
3: dec a;
4: inc b;
5: inc b;
6: inc b;
7: jnz a 3;
```

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, ..., `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, ...

Polecenia:

- `inc x;` $\equiv$ `x := x + 1;`
- `dec x;` $\equiv$ `x := x - 1;`
- `jnz x` i`;` $\equiv$ `if x <> 0 then goto` i`;`

Program: ciąg maksymalnie 21 instrukcji

Wynik: maksymalna wartość zmiennej po **zakończeniu** działania

0: **inc** a ;
1: **inc** a ;
2: **inc** a ;
3: **dec** a ;
4: **inc** b ;
5: **inc** b ;
6: **inc** b ;
7: **jnz** a 3;

wynik $= 9$

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, . . .

Polecenia:

- `inc x;` ≡ `x := x + 1;`
- `dec x;` ≡ `x := x - 1;`
- `jnz x` i`;` ≡ `if x <> 0 then goto` i`;`

Program: ciąg maksymalnie 21 instrukcji

Wynik: maksymalna wartość zmiennej po **zakończeniu** działania

Zadanie: napisać program o **możliwie dużym** wyniku

0 : **inc** a ;
1 : **inc** a ;
2 : **inc** a ;
3 : **dec** a ;
4 : **inc** b ;
5 : **inc** b ;
6 : **inc** b ;
7 : **jnz** a 3 ;

wynik $= 9$

Zajęcia praktyczno-techniczne

Zmienne: `a`, `b`, . . . , `z` (początkowo wszystkie równe 0)

Instrukcje numerowane 0, 1, . . .

Polecenia:

- `inc x;` $\equiv$ `x := x + 1;`
- `dec x;` $\equiv$ `x := x - 1;`
- `jnz x` i`;` $\equiv$ `if x <> 0 then goto` i`;`

Program: ciąg maksymalnie 21 instrukcji

Wynik: maksymalna wartość zmiennej

po <u>**zakończeniu**</u> działania

Zadanie: napisać program

o **możliwie dużym** wyniku

Rozwiązania: proszę wysyłać na

mskrzypczak@mimuw.edu.pl

0: **inc** a ;
1: **inc** a ;
2: **inc** a ;
3: **dec** a ;
4: **inc** b ;
5: **inc** b ;
6: **inc** b ;
7: **jnz** a 3 ;

wynik = 9

Podsumowanie

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Można je definiować jako wynik konkretnych **obliczeń**

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Można je definiować jako wynik konkretnych **obliczeń**

Np. $a \uparrow^n b$, n, $\ldots$

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Można je definiować jako wynik konkretnych **obliczeń**

Np. $a \uparrow^n b$, , …

Wielkość definiowanych liczb odzwierciedla **siłę** modelu obliczeń

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Można je definiować jako wynik konkretnych **obliczeń**

Np. $a \uparrow^n b$, n, …

Wielkość definiowanych liczb odzwierciedla **siłę** modelu obliczeń

Np. funkcje pierwotnie rekurencyjne vs. funkcja Ackermanna

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Można je definiować jako wynik konkretnych **obliczeń**

Np. $a \uparrow^n b$, , $\ldots$

Wielkość definiowanych liczb odzwierciedla **siłę** modelu obliczeń

Np. funkcje pierwotnie rekurencyjne vs. funkcja Ackermanna

Można też brać maksimum pewnych **klas obliczeń**, np: $\mathbf{BB}(n)$

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Można je definiować jako wynik konkretnych **obliczeń**

Np. $a \uparrow^n b$, , …

Wielkość definiowanych liczb odzwierciedla **siłę** modelu obliczeń

Np. funkcje pierwotnie rekurencyjne vs. funkcja Ackermanna

Można też brać maksimum pewnych **klas obliczeń**, np: $\mathbf{BB}(n)$

…ale wtedy dana liczba przestaje być obliczalna

Podsumowanie

Istnieją **naprawdę duże** liczby naturalne!

Można je definiować jako wynik konkretnych **obliczeń**

Np. $a \uparrow^n b$, n, …

Wielkość definiowanych liczb odzwierciedla **siłę** modelu obliczeń

Np. funkcje pierwotnie rekurencyjne vs. funkcja Ackermanna

Można też brać maksimum pewnych **klas obliczeń**, np: $\mathbf{BB}(n)$

… ale wtedy dana liczba przestaje być obliczalna

No i potrzeba trochę ostrożności, by uniknąć **paradoksu kłamcy**.