The (big) numbers game

Laboratoire Méthodes Formelles

Arnaud Golfouse

November 5th 2024

Why do we care?

Why do we care?

COMPETITION!

Why do we care?

The duel between Agustín Rayo and Adam Elga went like this:

- 1
- 111111111111111111111111
- $11!!!!!!!!!!!!!!!!!!!!!!$
- $\mathrm{BB}\left(10^{100}\right)$
- Busy Beaver hierarchy $\Rightarrow \mathrm{BB}_{\theta}\left(10^{100}\right)$
- Rayo's number

Outline

- Knuth's arrows
- Fast-growing hierarchy
- Busy beavers
- Rayo's number

Outline

- Knuth's arrows → Primitive recursive functions
- Fast-growing hierarchy → General recursion
- Busy beavers → Non-computable functions
- Rayo's number → Second-order logic

Knuth’s up-arrow notation

Iterate on operations

- addition: $100 + 100$
- multiplication: 100×100
- exponentiation: 100^{100}
- tower of exponentiations: $100^{\overbrace{\left(100^{100^{\cdot^{\cdot^{\cdot^{100}}}}}\right)}_{100 \text{ times}}}$

Knuth's up-arrow notation

$$a \uparrow b \triangleq a^b$$

$$a \uparrow\uparrow b \triangleq \underbrace{a \uparrow (a \uparrow (a \uparrow \ldots))}_{b \text{ copies of } a}$$

$$\ldots$$

$$a \uparrow^n b \triangleq \underbrace{a \uparrow^{n-1} (a \uparrow^{n-1} (a \uparrow^{n-1} \ldots))}_{b \text{ copies of } a}$$

Examples

$$\text{gogol} = 10^{100} = 10 \uparrow 100$$

$$\text{gogolplex} = 10^{10^{100}} < 10 \uparrow\uparrow 4$$

$$\underbrace{100^{100^{\cdot^{\cdot^{\cdot^{100}}}}}}_{100 \text{ times}} = 100 \uparrow\uparrow 100$$

Examples

$$\text{gogol} = 10^{100} = 10 \uparrow 100$$

$$\text{gogolplex} = 10^{10^{100}} < 10 \uparrow\uparrow 4$$

$$\underbrace{100^{100^{\cdot^{\cdot^{\cdot^{100}}}}}}_{100 \text{ times}} = 100 \uparrow\uparrow 100$$

$$3 \uparrow\uparrow\uparrow 3$$

Examples

$$\text{gogol} = 10^{100} = 10 \uparrow 100$$

$$\text{gogolplex} = 10^{10^{100}} < 10 \uparrow\uparrow 4$$

$$\underbrace{100^{100^{\cdot^{\cdot^{\cdot^{100}}}}}}_{100 \text{ times}} = 100 \uparrow\uparrow 100$$

$$3 \uparrow\uparrow\uparrow 3 = 3 \uparrow\uparrow (3 \uparrow\uparrow 3)$$

Examples

$$\text{gogol} = 10^{100} = 10 \uparrow 100$$

$$\text{gogolplex} = 10^{10^{100}} < 10 \uparrow\uparrow 4$$

$$\underbrace{100^{100^{\cdot^{\cdot^{\cdot^{100}}}}}}_{100 \text{ times}} = 100 \uparrow\uparrow 100$$

$$\begin{aligned} 3 \uparrow\uparrow\uparrow 3 &= 3 \uparrow\uparrow (3 \uparrow\uparrow 3) \\ &= 3 \uparrow\uparrow (3 \uparrow (3 \uparrow 3)) \end{aligned}$$

Examples

$$\text{gogol} = 10^{100} = 10 \uparrow 100$$

$$\text{gogolplex} = 10^{10^{100}} < 10 \uparrow\uparrow 4$$

$$\underbrace{100^{100^{\cdot^{\cdot^{\cdot^{100}}}}}}_{100 \text{ times}} = 100 \uparrow\uparrow 100$$

$$\begin{aligned} 3 \uparrow\uparrow\uparrow 3 &= 3 \uparrow\uparrow (3 \uparrow\uparrow 3) \\ &= 3 \uparrow\uparrow (3 \uparrow (3 \uparrow 3)) \\ &= 3 \uparrow\uparrow \left(3^{27}\right) \end{aligned}$$

Examples

$$\text{gogol} = 10^{100} = 10 \uparrow 100$$

$$\text{gogolplex} = 10^{10^{100}} < 10 \uparrow\uparrow 4$$

$$\underbrace{100^{100^{\cdot^{\cdot^{100}}}}}_{100 \text{ times}} = 100 \uparrow\uparrow 100$$

$$\begin{aligned} 3 \uparrow\uparrow\uparrow 3 &= 3 \uparrow\uparrow (3 \uparrow\uparrow 3) \\ &= 3 \uparrow\uparrow (3 \uparrow (3 \uparrow 3)) \\ &= 3 \uparrow\uparrow \left(3^{27}\right) \\ &= \underbrace{3^{3^{\cdot^{\cdot^{3}}}}}_{3^{27} \text{times}} \end{aligned}$$

Graham’s number

Graham's number

$$g_1 = 3 \uparrow\uparrow\uparrow\uparrow 3$$

Graham's number

$$g_2 = 3 \underbrace{\uparrow\uparrow \dots \uparrow}\; 3$$

$$g_1 = 3 \uparrow\uparrow\uparrow\uparrow 3$$

Graham's number

$$g_3 = \quad 3 \underbrace{\uparrow\uparrow \dots\dots\dots \uparrow}\ 3$$

$$g_2 = \quad 3 \underbrace{\uparrow\uparrow \dots \uparrow}\ 3$$

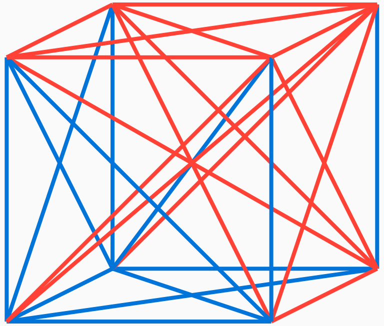
$$g_1 = \quad 3 \uparrow\uparrow\uparrow\uparrow 3$$

Graham's number

$$
\begin{aligned}
G = g_{64} &= 3 \underbrace{\uparrow\uparrow \cdots\cdots\cdots\cdots \uparrow}_{\displaystyle 3 \underbrace{\uparrow\uparrow \cdots\cdots\cdots \uparrow}_{\displaystyle \vdots} 3} 3 \\
g_3 &= 3 \underbrace{\uparrow\uparrow \cdots\cdots \uparrow} 3 \\
g_2 &= 3 \underbrace{\uparrow\uparrow \cdots \uparrow} 3 \\
g_1 &= 3 \uparrow\uparrow\uparrow\uparrow 3
\end{aligned}
$$

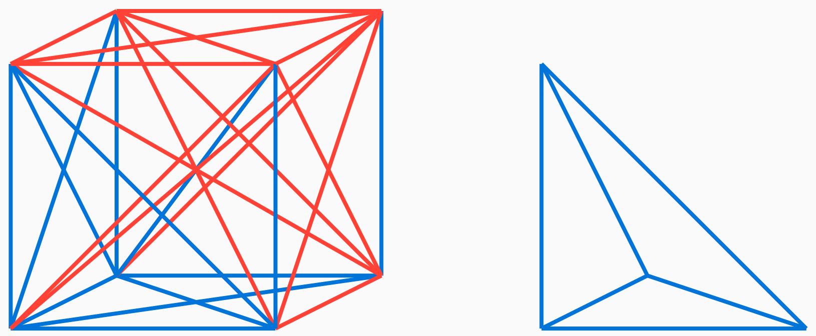
What is the use of g_{64}?

Take a coloring of an n-dimensional cube, and try to extract a single-colored subgraph of size 4:



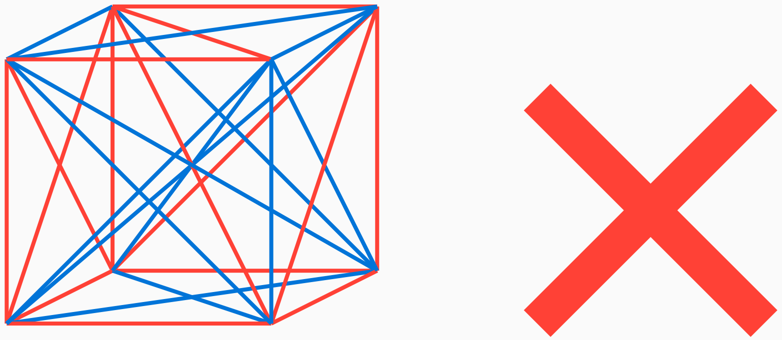
What is the use of g_{64}?

Take a coloring of an n-dimensional cube, and try to extract a single-colored subgraph of size 4:



What is the use of g_{64}?

Take a coloring of an n-dimensional cube, and try to extract a single-colored subgraph of size 4:



What is the use of g_{64}?

Find the smallest n, such that:

If you take the n-dimensional cube, and consider the complete graph of its vertices, any 2-coloring of the edges of this graph contains a complete subgraph of size 4 with a single color.

$$< n <$$

What is the use of g_{64}?

Find the smallest n, such that:

If you take the n-dimensional cube, and consider the complete graph of its vertices, any 2-coloring of the edges of this graph contains a complete subgraph of size 4 with a single color.

$$< n < g_{64}$$

What is the use of g_{64}?

Find the smallest n, such that:

If you take the n-dimensional cube, and consider the complete graph of its vertices, any 2-coloring of the edges of this graph contains a complete subgraph of size 4 with a single color.

$$13 < n < g_{64}$$

What is the use of g_{64}?

Find the smallest n, such that:

If you take the n-dimensional cube, and consider the complete graph of its vertices, any 2-coloring of the edges of this graph contains a complete subgraph of size 4 with a single color.

$$13 < n < 2 \uparrow\uparrow (2 \uparrow\uparrow 5138)$$

Fast-growing functions hierarchy

The fast-growing functions hierarchy

A family of functions $f_\alpha(n)$, where α is an *ordinal*.

Reminder on ordinals

- $0, 1, 2, 3, \ldots$
- $\omega = \{0, 1, 2, \ldots\}$ $\rightarrow$ 
- $\omega + 1 = \omega \cup \{\omega\}$ $\rightarrow$ 

Reminder on ordinals

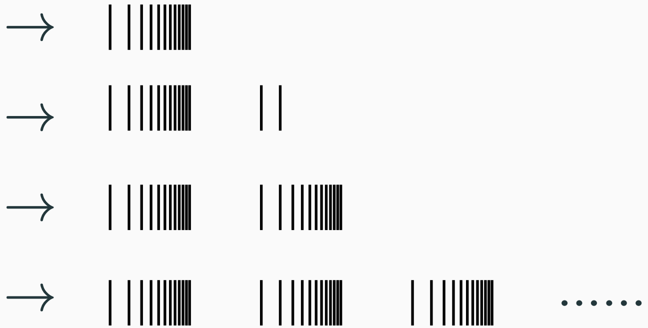
- $0, 1, 2, 3, \ldots$
- $\omega = \{0, 1, 2, \ldots\}$ $\rightarrow$ 
- $\omega + 2 = (\omega + 1) \cup \{\omega + 1\}$ $\rightarrow$ 

Reminder on ordinals

- $0, 1, 2, 3, \ldots$
- $\omega = \{0, 1, 2, \ldots\}$ $\rightarrow$
- $\omega + 2 = (\omega + 1) \cup \{\omega + 1\}$ $\rightarrow$
- $2\omega = \{0, 1, 2, \ldots, \omega, \omega + 1, \omega + 2, \ldots\}$ $\rightarrow$

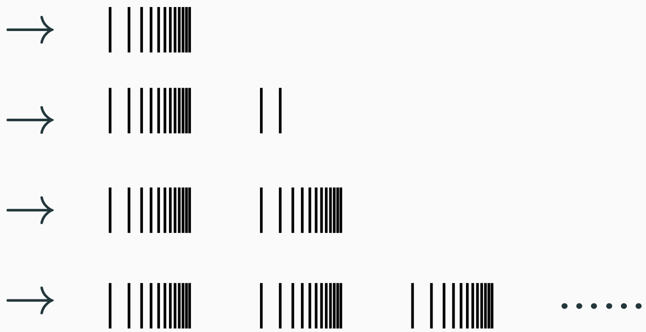
Reminder on ordinals

- $0, 1, 2, 3, \dots$
- $\omega = \{0, 1, 2, \dots\}$
- $\omega + 2 = (\omega + 1) \cup \{\omega + 1\}$
- $2\omega = \{0, 1, 2, \dots, \omega, \omega + 1, \omega + 2, \dots\}$
- $\omega^2 = \{\dots, 2\omega, 3\omega, \dots\}$
-
-



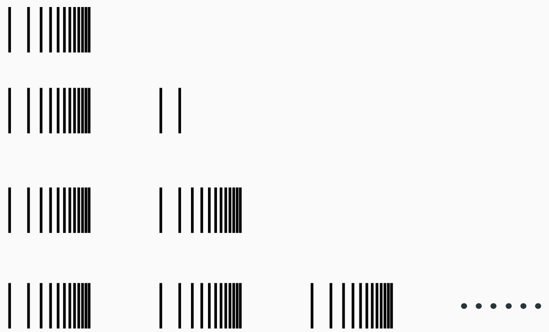
Reminder on ordinals

- $0, 1, 2, 3, \ldots$
- $\omega = \{0, 1, 2, \ldots\}$
- $\omega + 2 = (\omega + 1) \cup \{\omega + 1\}$
- $2\omega = \{0, 1, 2, \ldots, \omega, \omega + 1, \omega + 2, \ldots\}$
- $\omega^2 = \{\ldots, 2\omega, 3\omega, \ldots\}$
- $\omega^\omega = \{\ldots, \omega^2, \omega^3, \ldots\}$
-



Reminder on ordinals

- $0, 1, 2, 3, \ldots$
- $\omega = \{0, 1, 2, \ldots\}$ $\rightarrow$
- $\omega + 2 = (\omega + 1) \cup \{\omega + 1\}$ $\rightarrow$
- $2\omega = \{0, 1, 2, \ldots, \omega, \omega + 1, \omega + 2, \ldots\}$ $\rightarrow$
- $\omega^2 = \{\ldots, 2\omega, 3\omega, \ldots\}$ $\rightarrow$
- $\omega^\omega = \{\ldots, \omega^2, \omega^3, \ldots\}$
- $\omega^{\omega^\omega + 3} + \omega^2 + 5$



Some f_α functions

- $f_0(n) = n + 1$
- $f_1(n) =$
- $f_2(n) =$
- $f_\omega(n) =$
- $f_{\omega+1}(n) =$

Some f_α functions

- $f_0(n) = n + 1$
- $f_1(n) = (f_0)^n(n) = f_0(f_0(...f_0(n))) = 2n$
- $f_2(n) =$
- $f_\omega(n) =$
- $f_{\omega+1}(n) =$

Some f_α functions

- $f_0(n) = n + 1$
- $f_1(n) = (f_0)^n(n) = f_0(f_0(...f_0(n))) = 2n$
- $f_2(n) = f_1(f_1(...f_1(n))) = n \times 2^n$
- $f_\omega(n) =$
- $f_{\omega+1}(n) =$

Some f_α functions

- $f_0(n) = n + 1$
- $f_1(n) = (f_0)^n(n) = f_0(f_0(...f_0(n))) = 2n$
- $f_2(n) = f_1(f_1(...f_1(n))) = n \times 2^n$
- $f_\omega(n) = f_n(n)$
- $f_{\omega+1}(n) =$

Some f_α functions

- $f_0(n) = n + 1$
- $f_1(n) = (f_0)^n(n) = f_0(f_0(...f_0(n))) = 2n$
- $f_2(n) = f_1(f_1(...f_1(n))) = n \times 2^n$
- $f_\omega(n) = f_n(n)$
- $f_{\omega+1}(n) = f_\omega(f_\omega(...f_\omega(n)))$

Some f_α functions

- $$\begin{aligned} f_{2\omega}(n) &= f_{\omega+\omega}(n) \\ &= f_{\omega+n}(n) \end{aligned}$$
- $$\begin{aligned} f_{\omega^2}(n) &= f_{n\times\omega}(n) \\ &= f_{(n-1)\times\omega+n}(n) \end{aligned}$$
- $f_{\omega^\omega}(n) = f_{\omega^n}(n)$
- etc.

Definition of f_α

Let α be an ordinal (smaller than ε_0).

- $f_0(n) \triangleq n + 1$
- $f_{\alpha(n)}$:
  - If $\alpha = \alpha' + 1$, $f_{\alpha(n)} \triangleq (f_{\alpha'})^n(n)$
  - Else, $f_{\alpha(n)} \triangleq f_{\alpha[n]}(n)$, where →

- $\left(\omega^{\beta_0} + \omega^{\beta_1} + \ldots + \omega^{\beta_k}\right)[n] \triangleq$ $\omega^{\beta_0} + \omega^{\beta_1} + \ldots + \left(\omega^{\beta_k}[n]\right)$
- If $\beta = \beta' + 1$, $\omega^{\beta}[n] \triangleq n \times \omega^{\beta'}$
- Else, $\omega^{\beta}[n] \triangleq \omega^{\beta[n]}$

Examples

$$f_{\omega^2}(2)$$

Examples

$$
\begin{aligned}
f_{\omega^2}(2) &= f_{2\omega}(2) \\
&= f_{\omega+2}(2)
\end{aligned}
$$

Examples

$$
\begin{aligned}
f_{\omega^2}(2) &= f_{2\omega}(2) \\
&= f_{\omega+2}(2) \\
&= f_{\omega+1}(f_{\omega+1}(2)) \\
&= f_{\omega+1}(f_{\omega}(f_{\omega}(2)))
\end{aligned}
$$

Examples

$$
\begin{aligned}
f_{\omega^2}(2) &= f_{2\omega}(2) \\
&= f_{\omega+2}(2) \\
&= f_{\omega+1}(f_{\omega+1}(2)) \\
&= f_{\omega+1}(f_{\omega}(f_{\omega}(2))) \\
&= f_{\omega+1}(f_{\omega}(f_{2}(2)))
\end{aligned}
$$

Examples

$$
\begin{aligned}
f_{\omega^2}(2) &= f_{2\omega}(2) \\
&= f_{\omega+2}(2) \\
&= f_{\omega+1}(f_{\omega+1}(2)) \\
&= f_{\omega+1}(f_{\omega}(f_{\omega}(2))) \\
&= f_{\omega+1}(f_{\omega}(f_{2}(2))) \\
&= f_{\omega+1}(f_{\omega}(8))
\end{aligned}
$$

Examples

$$
\begin{aligned}
f_{\omega^2}(2) &= f_{2\omega}(2) \\
&= f_{\omega+2}(2) \\
&= f_{\omega+1}(f_{\omega+1}(2)) \\
&= f_{\omega+1}(f_{\omega}(f_{\omega}(2))) \\
&= f_{\omega+1}(f_{\omega}(f_{2}(2))) \\
&= f_{\omega+1}(f_{\omega}(8)) \\
&= f_{\omega+1}(f_{8}(8)) \\
&= \ldots
\end{aligned}
$$

Examples

$$f_\omega(n) = f_n(n)$$

Examples

$f_\omega(n) = f_n(n) \to \quad \approx$ Knuth's arrows !

Examples

$f_\omega(n) = f_n(n) \rightarrow \quad \approx$ Knuth's arrows !

$$f_{\omega+1}(n) = f_\omega(f_\omega(...f_\omega(n)))$$

Examples

$$f_\omega(n) = f_n(n) \rightarrow \quad \approx \text{Knuth's arrows !}$$

$$f_{\omega+1}(n) = f_\omega(f_\omega(...f_\omega(n))) \rightarrow \text{Graham's number !}$$

$$f_{\varepsilon_0}(n) = f_{\underbrace{\omega^{\omega^{\cdots^{\omega}}}}_{n \text{ times}}}(n)$$

Busy beavers and oracles

Busy beavers

The second-to-last of Rayo's propositions was $\mathrm{BB}\left(10^{100}\right)$: what is that?

Busy beavers

Take all turing machines that:

- Have the alphabet $\{0, 1\}$
- Have n or less states
- Run on a originally all 0 tape
- Terminate

Busy beavers

Take all turing machines that:

- Have the alphabet $\{0, 1\}$
- Have n or less states
- Run on a originally all 0 tape
- Terminate

Then $\mathrm{BB}(n)$ is the maximum numbers of 1s at the end of the execution of such a machine.

Busy beavers

Take all turing machines that:

- Have the alphabet $\{0, 1\}$
- Have n or less states
- Run on a originally all 0 tape
- Terminate

Then $\mathrm{BB}(n)$ is the maximum numbers of 1s at the end of the execution of such a machine.

→ This already beats our previous solutions!

Oracles

Now add an *oracle* : a black box, that says whether a normal Turing machine terminates or not.

With those "second-order" Turing machines, we can define $\mathrm{BB}_2(n)$.

Oracles

Now add an *oracle* : a black box, that says whether a normal Turing machine terminates or not.

With those “second-order” Turing machines, we can define $\mathrm{BB}_2(n)$.

Then $\mathrm{BB}_3(n)$, $\mathrm{BB}_4(n)$, $\mathrm{BB}_5(n)$, ...

Oracles

Now add an *oracle* : a black box, that says whether a normal Turing machine terminates or not.

With those “second-order” Turing machines, we can define $\mathrm{BB}_2(n)$.

Then $\mathrm{BB}_3(n)$, $\mathrm{BB}_4(n)$, $\mathrm{BB}_5(n)$, ...

Then $\mathrm{BB}_\omega(n)$...

Oracles

Now add an *oracle* : a black box, that says whether a normal Turing machine terminates or not.

With those “second-order” Turing machines, we can define $\mathrm{BB}_2(n)$.

Then $\mathrm{BB}_3(n)$, $\mathrm{BB}_4(n)$, $\mathrm{BB}_5(n)$, ...

Then $\mathrm{BB}_\omega(n)$...

Same as before, we need to stop eventually: Adam Elga stopped at $\mathrm{BB}_\theta\left(10^{100}\right)$.

Winner?

Winner

So, can we even go beyond that in any meaningful way?

Winner

So, can we even go beyond that in any meaningful way? *yes!*

Winner

We generalized over the notion of computation : it’s time to generalize over logic itself!

Winner

We generalized over the notion of computation : it's time to generalize over logic itself!

Rayo's number is the biggest number that can be defined with a first-order formula of size 10^{100}.

The formula

If we have a function $[\cdot] : \text{formula} \to \mathbb{N}$, define the second-order formula $\mathrm{Sat}([\varphi], s)$ to mean

$$\begin{aligned}
&\text{for all } R, \{\forall \psi, \forall t \text{ a variable assignement},\\
&\qquad R([\psi], t) \leftrightarrow \big(\big(\psi = \text{"}x_i \in x_j\text{"} \wedge t(x_i) \in t\big(x_j\big)\big) \vee \\
&\qquad\qquad \big(\psi = \text{"}x_i = x_j\text{"} \wedge t(x_i) = t\big(x_j\big)\big) \vee \\
&\qquad\qquad (\psi = \text{"}\neg\theta\text{"} \wedge \neg R([\theta], t)) \vee \\
&\qquad\qquad (\psi = \text{"}\theta_1 \wedge \theta_2\text{"} \wedge R([\theta_1], t) \wedge R([\theta_2], t)) \vee \\
&\qquad\qquad (\psi = \text{"}\exists x_i \theta\text{"} \wedge \exists X, R([\theta], t[x_i \leftarrow X]))\big)\\
&\quad\} \to R([\varphi], s)
\end{aligned}$$

The formula

Then Rayo's number is

$$
\begin{aligned}
R := \max\Big(\Big\{m \in \mathbb{N} \mid\ & \exists \varphi \in \mathrm{arity}(1), \\
& |\varphi| \leq 10^{100} \\
& \exists s, \mathrm{Sat}([\varphi], s) \wedge s(x_1) = m \\
& \forall t, \mathrm{Sat}([\varphi], t) \rightarrow t(x_1) = m \\
\Big\}\Big) &
\end{aligned}
$$

Thank you!