Ackermann’s Function

around 1900….

World Exhibition in Paris

around the year 1900

Eiffel Tower completed

World Exhibition in Paris

around the year 1900

Eiffel Tower completed

David Hilbert

- 38 years old
- Professor at Göttingen

around the year 1900

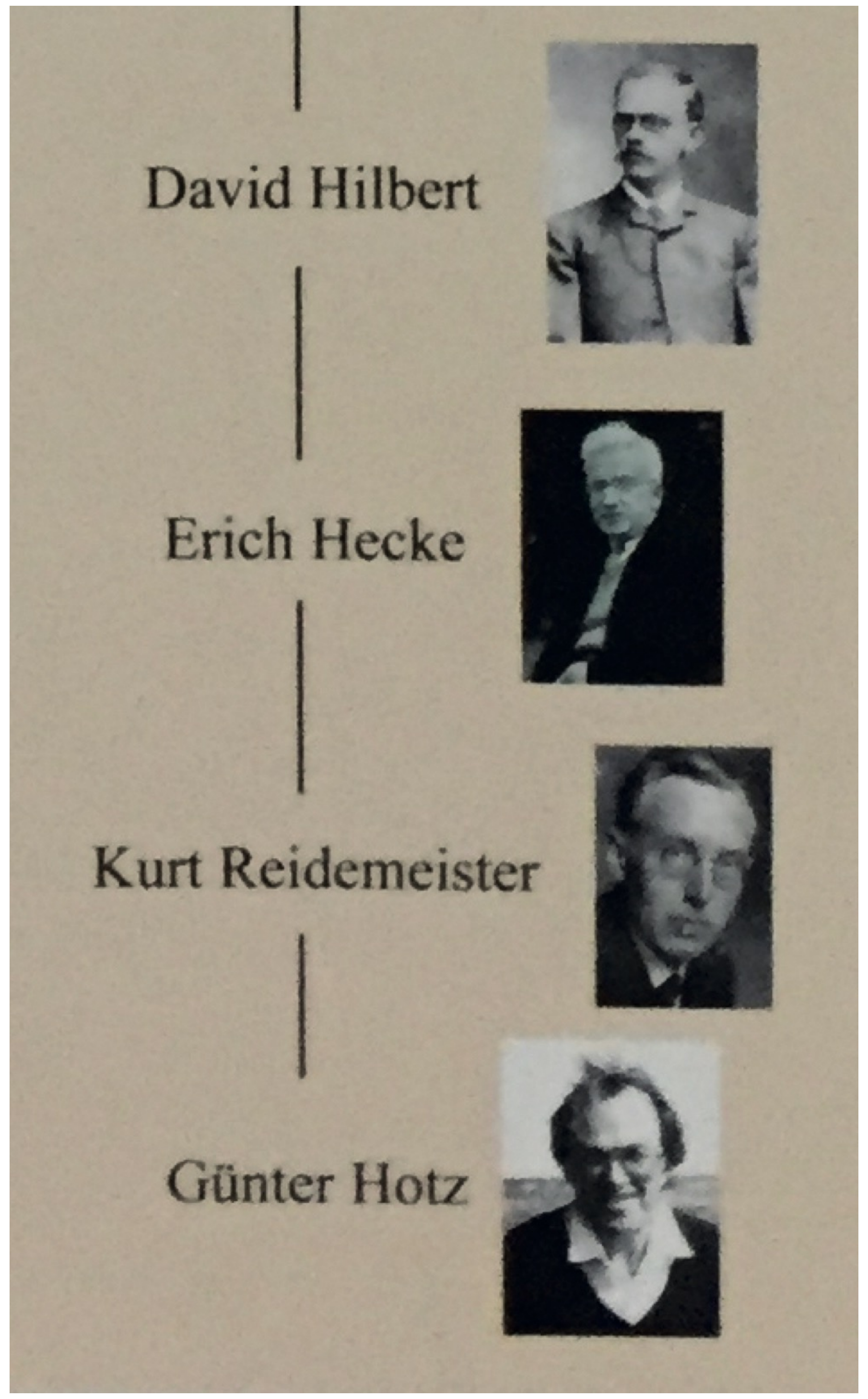
World Exhibition in Paris

Eiffel Tower completed

David Hilbert

- 38 years old
- Professor at Göttingen

some descendants



around the year 1900

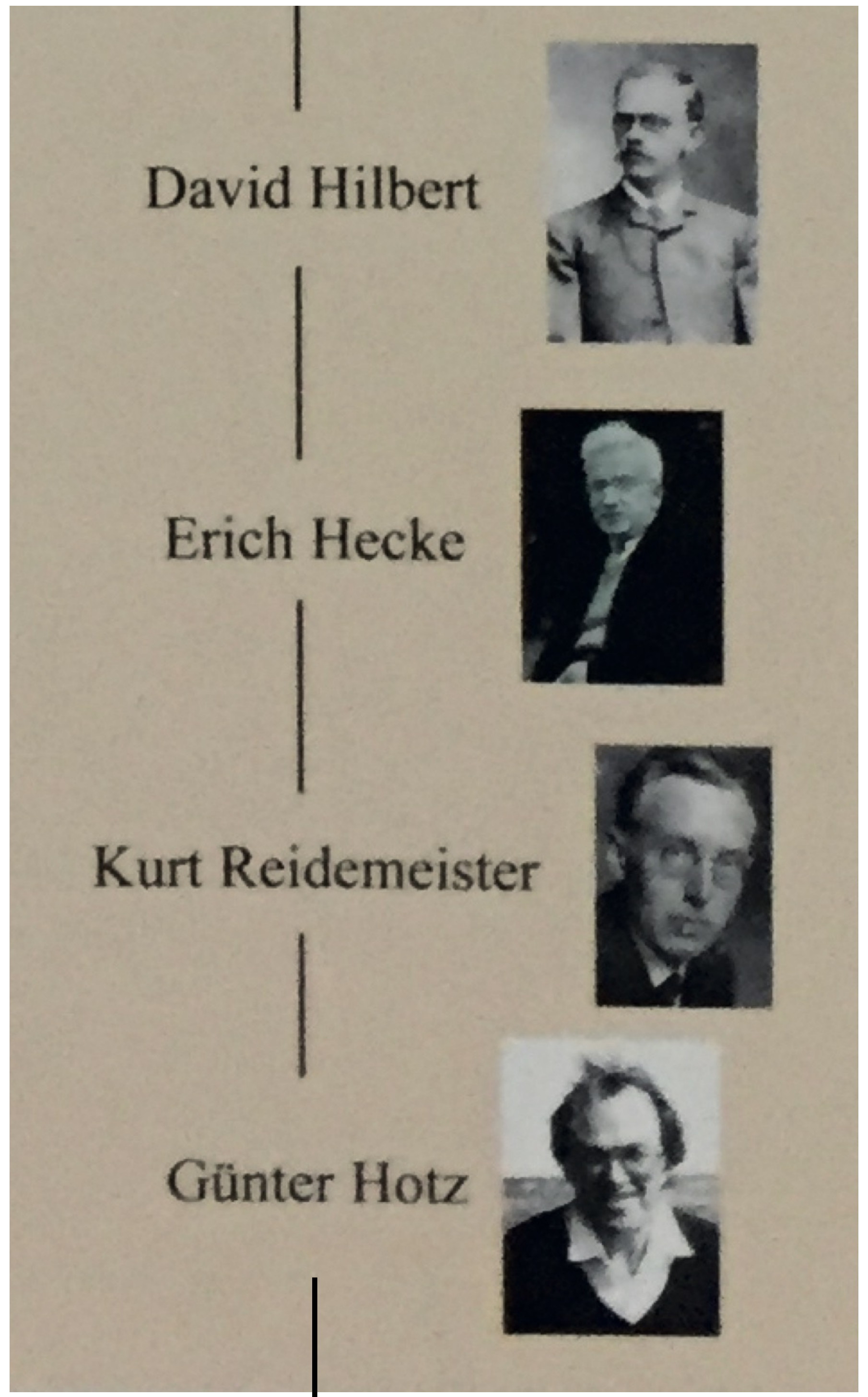
World Exhibition in Paris

Eiffel Tower completed

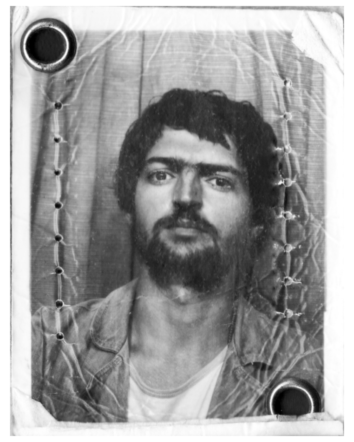
David Hilbert

- 38 years old
- Professor at Göttingen
- my doctoral grand-grand-grand-father

some descendants



Wolfgang Paul



around the year 1900

World Exhibition in Paris

Eiffel Tower completed

David Hilbert

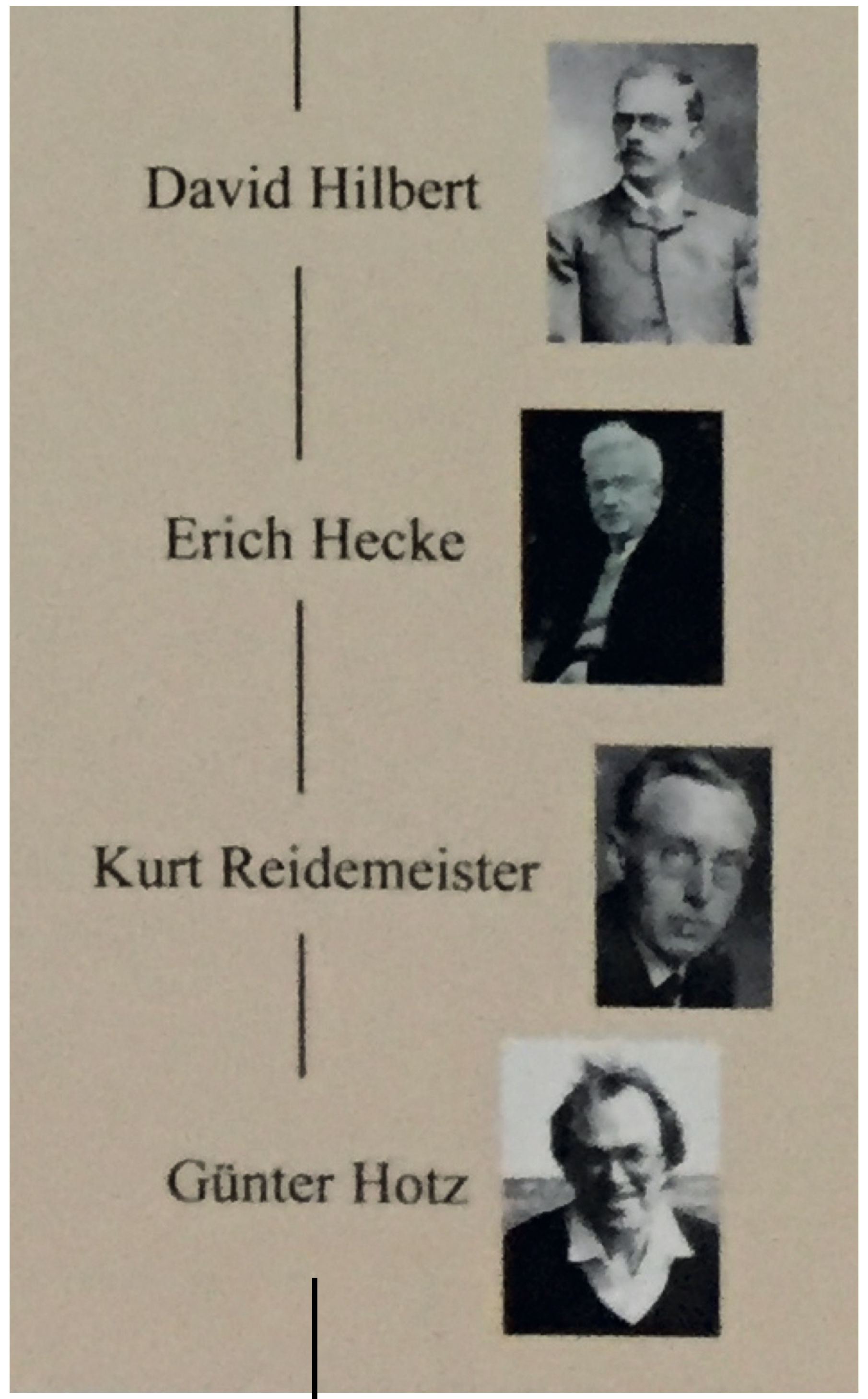
- 38 years old
- Professor at Göttingen
- my doctoral grand-grand-grand-father

now

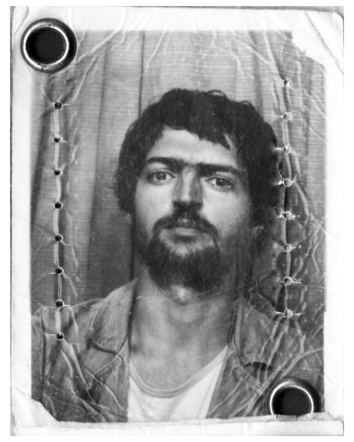
if you think, that I have a big ego…

it is **NOTHING** compared to my ancestor

some descendants



Wolfgang Paul



around the year 1900

World Exhibition in Paris

Eiffel Tower completed

David Hilbert

- 38 years old
- Professor at Göttingen
- my doctoral grand-grand-grand-father

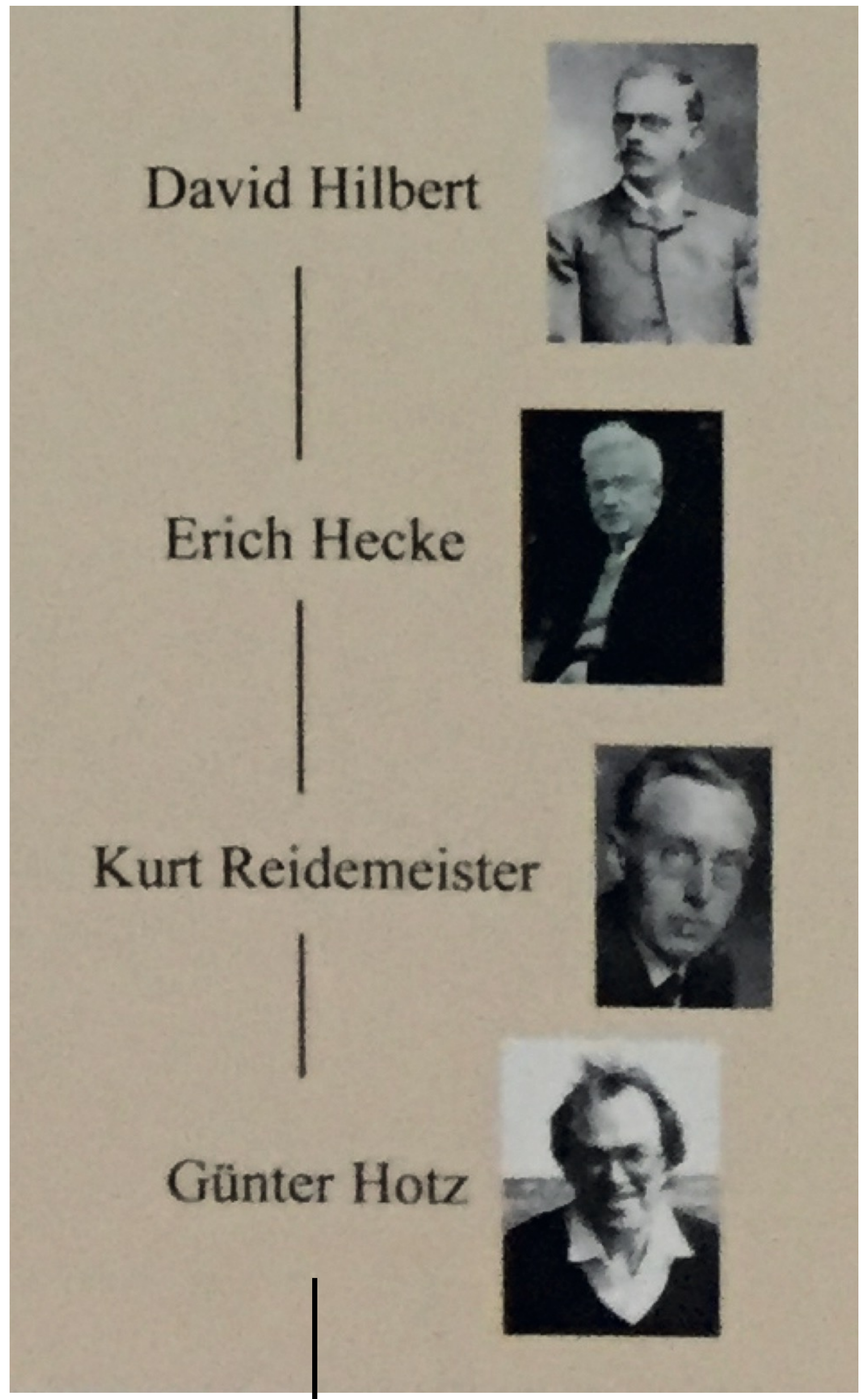
now

if you think, that I have a big ego…

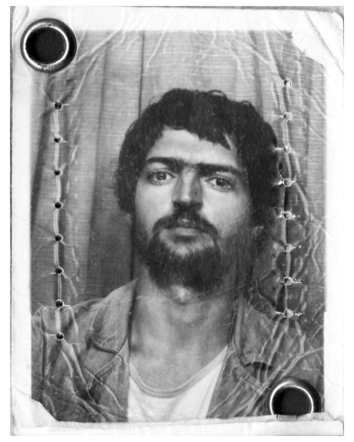
it is **NOTHING** compared to my ancestor

you think that is impossible….?

some descendants



Wolfgang Paul



Hilbert Conjecture and Hilbert Program

Conjecture: Every mathematical problem is solvable

Hilbert Conjecture and Hilbert Program

Conjecture: Every mathematical problem is solvable

- this was **not** meant as a topic of philosophical discourse
- he **had a plan** how to prove this

Hilbert Conjecture and Hilbert Program

Conjecture: Every mathematical problem is solvable

- this was **not** meant as a topic of philosophical discourse
- he **had a plan** how to prove this

Hilbert Program

- formally define the language L of mathematical predicates and statements. Today: *predicate calculus*.
- formally define, what is a proof: today elementary number theory (with $=, +, \cdot$) and Zermelo-Fraenkel set theory.
- define what is a computing method and the functions computed by them. Today 'computable functions', but at the time there were no computers.
- show that there is a (possible extremely complex and slow) computing methodes which decides for statements $s \in L$ whether it is provable or not.

Hilbert Conjecture and Hilbert Program

Conjecture: Every mathematical problem is solvable

- this was **not** meant as a topic of philosophical discourse
- he **had a plan** how to prove this

Hilbert Program

- formally define the language L of mathematical predicates and statements. Today: *predicate calculus*.
- formally define, what is a proof: today elementary number theory (with $=, +, \cdot$) and Zermelo-Fraenkel set theory.
- define what is a computing method and the functions computed by them. Today 'computable functions', but at the time there were no computers.
- show that there is a (possible extremely complex and slow) computing methodes which decides for statements $s \in L$ whether it is provable or not.

Goedel

- showed around 1931: such a computing method does not exists (incompleteness theorem).
- proof in I2TCS (introduction to theoretical computer science)

primitive recursive functions

conjecture at the time: this are the computable functions

Inductive definition of a set PR of computable functions:

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

1. constant functions $c_s^r \in PR$ where

$$c_s^r : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$c_s^r(x) = s \ , \ s \in \mathbb{N}_0$$

2. projections $P_i^r \in PR$ where

$$P_i^r(x) : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$P_i^r(x) = x_i$$

3. successor function $S \in PR$

$$s : \mathbb{N}_0 \to \mathbb{N}_0$$

$$S(x) = x + 1$$

primitive recursive functions

conjecture at the time: this are the computable functions

Inductive definition of a set PR of computable functions:

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

1. constant functions $c_s^r \in PR$ where

$$c_s^r : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$c_s^r(x) = s \ , \ s \in \mathbb{N}_0$$

2. projections $P_i^r \in PR$ where

$$P_i^r(x) : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$P_i^r(x) = x_i$$

3. successor function $S \in PR$

$$s : \mathbb{N}_0 \to \mathbb{N}_0$$

$$S(x) = x + 1$$

4. substitution. If the following function are all in PR

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also $h \in PR$ where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

5. primitive recursion. If the following functions are in PR

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also $f \in PR$ where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

primitive recursive functions: examples

conjecture at the time: this are the computable functions

- addition

$$
\begin{aligned}
f(0,x) &= x \\
f(n+1,x) &= S(f(n,x))
\end{aligned}
$$

primitive recursive functions: examples

conjecture at the time: this are the computable functions

- addition

$$
\begin{aligned}
f(0,x) &= x \\
f(n+1,x) &= S(f(n,x))
\end{aligned}
$$

- multiplication

$$
\begin{aligned}
g(0,x) &= 0 \\
g(n+1,x) &= f(x,g(n,x))
\end{aligned}
$$

primitive recursive functions: examples

conjecture at the time: this are the computable functions

- addition

$$\begin{aligned} f(0,x) &= x \\ f(n+1,x) &= S(f(n,x)) \end{aligned}$$

- multiplication

$$\begin{aligned} g(0,x) &= 0 \\ g(n+1,x) &= f(x,g(n,x)) \end{aligned}$$

- exponentiation

$$\begin{aligned} h(0,x) &= 1 \\ h(n+1,x) &= g(n,h(n,x)) \end{aligned}$$

primitive recursive functions: examples

conjecture at the time: this are the computable functions

- addition

$$\begin{aligned} f(0,x) &= x \\ f(n+1,x) &= S(f(n,x)) \end{aligned}$$

- multiplication

$$\begin{aligned} g(0,x) &= 0 \\ g(n+1,x) &= f(x,g(n,x)) \end{aligned}$$

- exponentiation

$$\begin{aligned} h(0,x) &= 1 \\ h(n+1,x) &= g(n,h(n,x)) \end{aligned}$$

- predecessor in $\mathbb{N}_0$

$$\begin{aligned} p(0) &= 0 \\ p(n+1) &= n \end{aligned}$$

Ackermann function

Ackermann 1926

Lemma 1. *There exists a total computable function, which is not promitive recursive*

- addition

$$\begin{aligned} f(0,x) &= x \\ f(n+1,x) &= S(f(n,x)) \end{aligned}$$

- multiplication

$$\begin{aligned} g(0,x) &= 0 \\ g(n+1,x) &= f(x,g(n,x)) \end{aligned}$$

- exponentiation

$$\begin{aligned} h(0,x) &= 1 \\ h(n+1,x) &= g(n,h(n,x)) \end{aligned}$$

- predecessor in $\mathbb{N}_0$

$$\begin{aligned} p(0) &= 0 \\ p(n+1) &= n \end{aligned}$$

Ackermann function

Ackermann 1926

Lemma 1. *There exists a total computable function, which is not promitive recursive*

- addition

$$\begin{aligned} f(0,x) &= x \\ f(n+1,x) &= S(f(n,x)) \end{aligned}$$

- multiplication

$$\begin{aligned} g(0,x) &= 0 \\ g(n+1,x) &= f(x,g(n,x)) \end{aligned}$$

- exponentiation

$$\begin{aligned} h(0,x) &= 1 \\ h(n+1,x) &= g(n,h(n,x)) \end{aligned}$$

- predecessor in $\mathbb{N}_0$

$$\begin{aligned} p(0) &= 0 \\ p(n+1) &= n \end{aligned}$$

$$\begin{aligned} \phi(a,b,0) &= a+b \\ \phi(a,0,n+1) &= \beta(a,b) \\ \phi(a,b+1,n+1) &= \phi(a,\phi(a,b,n+1),n) \end{aligned}$$

where

$$\beta(a,n) = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ a & n>1 \end{cases}$$

Then

$$\begin{aligned} \phi(a,b,0) &= a+b \\ \phi(a,b,1) &= a \cdot b \\ \phi(a,b,2) &= a^b \\ &\ldots \end{aligned}$$

Ackermann function

- addition

$$\begin{aligned} f(0,x) &= x \\ f(n+1,x) &= S(f(n,x)) \end{aligned}$$

- multiplication

$$\begin{aligned} g(0,x) &= 0 \\ g(n+1,x) &= f(x,g(n,x)) \end{aligned}$$

- exponentiation

$$\begin{aligned} h(0,x) &= 1 \\ h(n+1,x) &= g(n,h(n,x)) \end{aligned}$$

- predecessor in $\mathbb{N}_0$

$$\begin{aligned} p(0) &= 0 \\ p(n+1) &= n \end{aligned}$$

Ackermann 1926

Lemma 1. *There exists a total computable function, which is not promitive recursive*

$$\begin{aligned} \phi(a,b,0) &= a+b \\ \phi(a,0,n+1) &= \beta(a,b) \\ \phi(a,b+1,n+1) &= \phi(a,\phi(a,b,n+1),n) \end{aligned}$$

where

$$\beta(a,n) = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ a & n>1 \end{cases}$$

Then

$$\begin{aligned} \phi(a,b,0) &= a+b \\ \phi(a,b,1) &= a \cdot b \\ \phi(a,b,2) &= a^b \\ &\dots \end{aligned}$$

Growth argument:

- any function $f(n)$ defined by k applications of rules for PR grows at most as $\phi(n,n,k)$
- $\phi(n,n,n)$ grows faster than any on these functions

a close relative of the Ackermann function

Iterating i times function f:

$$\begin{aligned} f^{(0)}(j) &= j \\ f^{(i+1)}(j) &= f(f^{(i)}(j)) \end{aligned}$$

$$A_k(j) = \begin{cases} j+1 & k=0 \\ A_{k-1}^{(j+1)}(j) & \end{cases}$$

a close relative of the Ackermann function

Iterating i times function f:

$$\begin{aligned} f^{(0)}(j) &= j \\ f^{(i+1)}(j) &= f(f^{(i)}(j)) \end{aligned}$$

$$A_k(j) = \begin{cases} j+1 & k=0 \\ A_{k-1}^{(j+1)}(j) & \end{cases}$$

$$A_0(j) = j+1 = S(j)$$

a close relative of the Ackermann function

Iterating i times function f:

$$\begin{aligned} f^{(0)}(j) &= j \\ f^{(i+1)}(j) &= f(f^{(i)}(j)) \end{aligned}$$

$$A_k(j) = \begin{cases} j+1 & k=0 \\ A_{k-1}^{(j+1)}(j) & \end{cases}$$

$$A_0(j) = j+1 = S(j)$$

Lemma 2.

$$A_1(j) = 2j+1$$

$$S^{(i)}(j) = j + (i \cancel{+1}) \quad \text{(induction on } i\text{)}$$

$$\begin{aligned} A_1(j) &= S^{(j+1)}(j) \\ &= j + (j+1) \end{aligned}$$

a close relative of the Ackermann function

Iterating i times function f:

$$\begin{aligned} f^{(0)}(j) &= j \\ f^{(i+1)}(j) &= f(f^{(i)}(j)) \end{aligned}$$

$$A_k(j) = \begin{cases} j+1 & k=0 \\ A_{k-1}^{(j+1)}(j) \end{cases}$$

$$A_0(j) = j+1 = S(j)$$

Lemma 2.

$$A_1(j) = 2j+1$$

$$S^{(i)}(j) = j + (i \cancel{+1}) \quad \text{(induction on } i\text{)}$$

$$\begin{aligned} A_1(j) &= S^{(j+1)}(j) \\ &= j + (j+1) \end{aligned}$$

Lemma 3.

$$A_2(j) = 2^{j+1}(j+1) - 1$$

Proof. exercise

a close relative of the Ackermann function

Iterating i times function f:

$$
\begin{aligned}
f^{(0)}(j) &= j \\
f^{(i+1)}(j) &= f(f^{(i)}(j))
\end{aligned}
$$

$$
A_k(j) = \begin{cases} j+1 & k=0 \\ A_{k-1}^{(j+1)}(j) \end{cases}
$$

$$A_0(j) = j+1 = S(j)$$

Lemma 2.

$$A_1(j) = 2j+1$$

$$S^{(i)}(j) = j + (i \cancel{+1}) \quad \text{(induction on } i\text{)}$$

$$
\begin{aligned}
A_1(j) &= S^{(j+1)}(j) \\
&= j+(j+1)
\end{aligned}
$$

Lemma 3.

$$A_2(j) = 2^{j+1}(j+1)-1$$

Proof. exercise

$$
\begin{aligned}
A_0(1) &= 2 \\
A_1(1) &= 3 \quad \text{(lemma 2)} \\
A_2(1) &= 7 \quad \text{(lemma 3)}
\end{aligned}
$$

a close relative of the Ackermann function

Iterating i times function f:

$$\begin{aligned} f^{(0)}(j) &= j \\ f^{(i+1)}(j) &= f(f^{(i)}(j)) \end{aligned}$$

$$A_k(j) = \begin{cases} j+1 & k=0 \\ A_{k-1}^{(j+1)}(j) \end{cases}$$

$$A_0(j) = j+1 = S(j)$$

Lemma 2.

$$A_1(j) = 2j+1$$

$$S^{(i)}(j) = j + (i \cancel{+1}) \quad \text{(induction on } i\text{)}$$

$$\begin{aligned} A_1(j) &= S^{(j+1)}(j) \\ &= j+(j+1) \end{aligned}$$

Lemma 3.

$$A_2(j) = 2^{j+1}(j+1) - 1$$

Proof. exercise

$$\begin{aligned} A_0(1) &= 2 \\ A_1(1) &= 3 \quad \text{(lemma 2)} \\ A_2(1) &= 7 \quad \text{(lemma 3)} \end{aligned}$$

$$\begin{aligned} A_3(1) &= A_2^2(1) \\ &= A_2(A_2(1)) \\ &= A_2(7) \\ &= A_2(7) \\ &= \cancel{2^6 \cdot 8 - 1} \quad 2^8 \cdot 8 - 1 \\ &= 2^{11} - 1 \\ &= 2047 \end{aligned}$$

a close relative of the Ackermann function

Iterating i times function f:

$$
\begin{aligned}
f^{(0)}(j) &= j \\
f^{(i+1)}(j) &= f(f^{(i)}(j))
\end{aligned}
$$

$$
A_k(j) = \begin{cases} j+1 & k=0 \\ A_{k-1}^{(j+1)}(j) \end{cases}
$$

$$A_0(j) = j+1 = S(j)$$

Lemma 2.

$$A_1(j) = 2j+1$$

$$S^{(i)}(j) = j + (i \cancel{+1}) \quad \text{(induction on } i\text{)}$$

$$
\begin{aligned}
A_1(j) &= S^{(j+1)}(j) \\
&= j + (j+1)
\end{aligned}
$$

Lemma 3.

$$A_2(j) = 2^{j+1}(j+1) - 1$$

Proof. exercise

$$
\begin{aligned}
A_0(1) &= 2 \\
A_1(1) &= 3 \quad \text{(lemma 2)} \\
A_2(1) &= 7 \quad \text{(lemma 3)}
\end{aligned}
$$

$$
\begin{aligned}
A_3(1) &= A_2^2(1) \\
&= A_2(A_2(1)) \\
&= A_2(7) \\
&= A_2(7) \\
&= \cancel{2^6 \cdot 8 - 1} \quad 2^8 \cdot 8 - 1 \\
&= 2^{11} - 1 \\
&= 2047
\end{aligned}
$$

$$
\begin{aligned}
A_4(1) &= A_3^{(2)}(1) \\
&= A_3(A_3(1)) \\
&= A_3(2047) \\
&= A_2^{(2048)}(2047) \\
&>> A_2(2047) \\
&= \cancel{2^{2048}(2047)} \quad 2^{2048} \cdot 2048 - 1 \\
&> 2^{2048} \\
&= (2^4)^{512} \\
&>> 10^{18} \quad \text{estimated number of atoms in universe}
\end{aligned}
$$

inverse of the Ackermann function

$$
\begin{aligned}
A_0(1) &= 2 \\
A_1(1) &= 3 \quad \text{(lemma 2)} \\
A_2(1) &= 7 \quad \text{(lemma 3)}
\end{aligned}
$$

$$
\begin{aligned}
A_3(1) &= A_2^2(1) \\
&= A_2(A_2(1)) \\
&= A_2(7) \\
&= A_2(7) \\
&= \cancel{2^6 \cdot 8 - 1} \quad 2^8 \cdot 8 - 1 \\
&= 2^{11} - 1 \\
&= 2047
\end{aligned}
$$

$$
\begin{aligned}
A_4(1) &= A_3^{(2)}(1) \\
&= A_3(A_3(1)) \\
&= A_3(2047) \\
&= A_2^{(2048)}(2047) \\
&>> A_2(2047) \\
&= 2^{\cancel{2048}}\cancel{(2047})) \quad 2^{2048} \cdot 2048 - 1 \\
&> 2^{2048} \\
&= (2^4)^{512} \\
&>> 10^{18} \quad \text{estimated number of atoms in universe}
\end{aligned}
$$

$$\alpha(n) = \min\{k \mid A_k(1) \geq n\}$$

$$
\alpha(n) = \begin{cases}
0 & 0 \leq n \leq 2 \\
1 & n = 3 \\
2 & 4 \leq n \leq 7 \\
3 & 8 \leq n \leq 2047 \\
4 & 2048 \leq n \leq A_4(1)
\end{cases}
$$

inverse of the Ackermann function

$$
\begin{aligned}
A_0(1) &= 2 \\
A_1(1) &= 3 \quad \text{(lemma 2)} \\
A_2(1) &= 7 \quad \text{(lemma 3)}
\end{aligned}
$$

$$
\begin{aligned}
A_3(1) &= A_2^2(1) \\
&= A_2(A_2(1)) \\
&= A_2(7) \\
&= A_2(7) \\
&= \cancel{2^6 \cdot 8 - 1} \quad 2^8 \cdot 8 - 1 \\
&= 2^{11} - 1 \\
&= 2047
\end{aligned}
$$

$$
\begin{aligned}
A_4(1) &= A_3^{(2)}(1) \\
&= A_3(A_3(1)) \\
&= A_3(2047) \\
&= A_2^{(2048)}(2047) \\
&>> A_2(2047) \\
&= 2^{\cancel{2048}(2047)} \quad 2^{2048} \cdot 2048 - 1 \\
&> 2^{2048} \\
&= (2^4)^{512} \\
&>> 10^{18} \quad \text{estimated number of atoms in universe}
\end{aligned}
$$

$$\alpha(n) = \min\{k \mid A_k(1) \geq n\}$$

$$
\alpha(n) = \begin{cases}
0 & 0 \leq n \leq 2 \\
1 & n = 3 \\
2 & 4 \leq n \leq 7 \\
3 & 8 \leq n \leq 2047 \\
4 & 2048 \leq n \leq A_4(1)
\end{cases}
$$

$\boxed{\text{run time of union-find algorithm with path compression: } O(n + m \cdot \alpha(n))}$

inverse of the Ackermann function

$$
\begin{aligned}
A_0(1) &= 2 \\
A_1(1) &= 3 \quad \text{(lemma 2)} \\
A_2(1) &= 7 \quad \text{(lemma 3)}
\end{aligned}
$$

$$
\begin{aligned}
A_3(1) &= A_2^2(1) \\
&= A_2(A_2(1)) \\
&= A_2(7) \\
&= A_2(7) \\
&= \cancel{2^6 \cdot 8 - 1} \quad 2^8 \cdot 8 - 1 \\
&= 2^{11} - 1 \\
&= 2047
\end{aligned}
$$

$$
\begin{aligned}
A_4(1) &= A_3^{(2)}(1) \\
&= A_3(A_3(1)) \\
&= A_3(2047) \\
&= A_2^{(2048)}(2047) \\
&>> A_2(2047) \\
&= \cancel{2^{2048}(2047)} \quad 2^{2048} \cdot 2048 - 1 \\
&> 2^{2048} \\
&= (2^4)^{512} \\
&>> 10^{18} \quad \text{estimated number of atoms in universe}
\end{aligned}
$$

$$\alpha(n) = \min\{k \mid A_k(1) \geq n\}$$

$$
\alpha(n) = \begin{cases}
0 & 0 \leq n \leq 2 \\
1 & n = 3 \\
2 & 4 \leq n \leq 7 \\
3 & 8 \leq n \leq 2047 \\
4 & 2048 \leq n \leq A_4(1)
\end{cases}
$$

run time of union-find algorithm with path compression: $O(n + m \cdot \alpha(n))$

- all instincts say/scream: **this cannot possibly be true**
- however, there is proof…