Bizonyítás és programozás

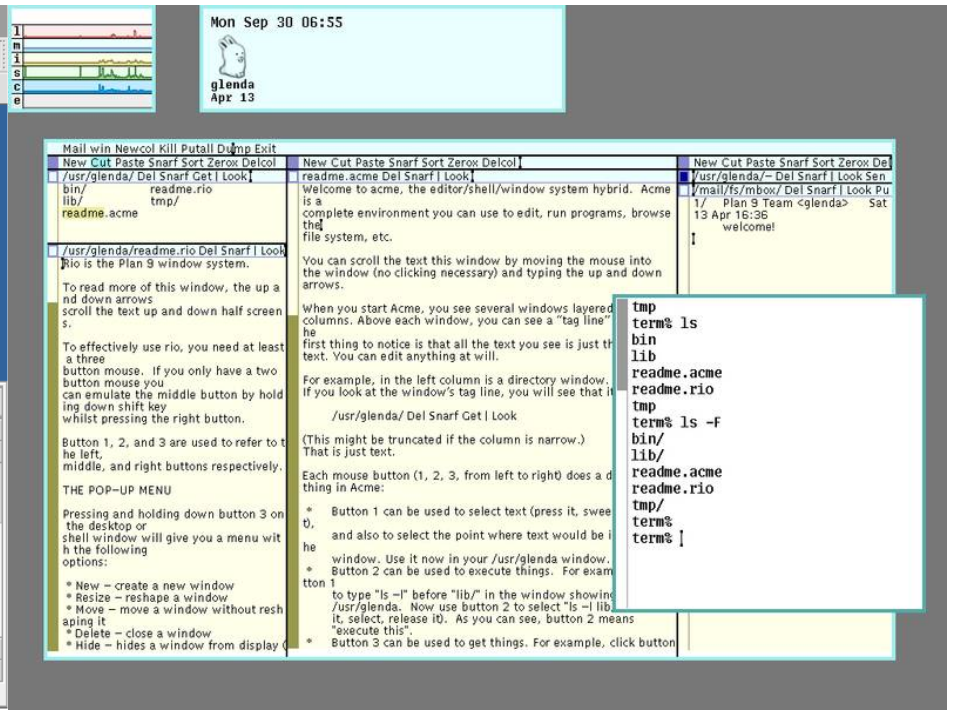
Kaposi Ambrus

Eötvös Loránd Tudományegyetem
Informatikai Kar

Pannonhalmi Bencés Gimnázium
2017. november 24.

A tökéletes operációs rendszer (i)

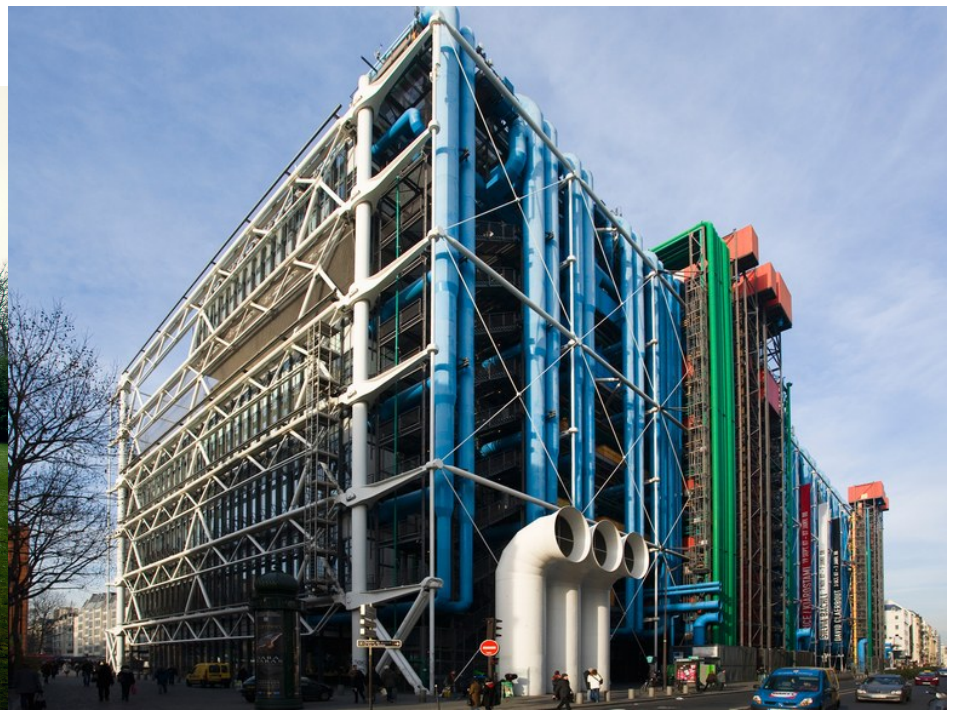
BeOS



Plan9

A tökéletes operációs rendszer (ii)

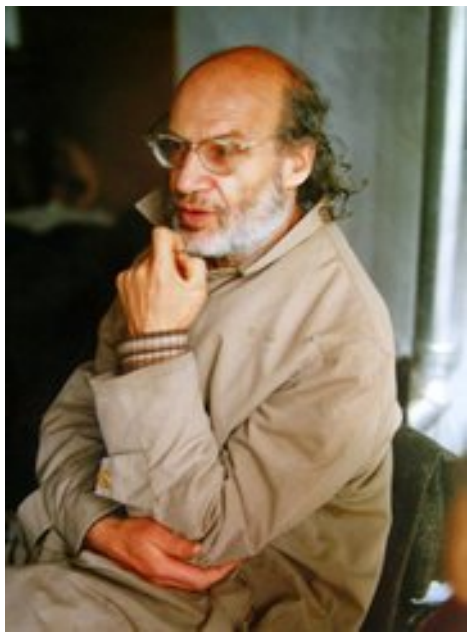
Villa Savoye



Centre Pompidou

Logisták

a program helyesen működjön
modularitás, absztrakció
új rendszer
nyelvész
forma



Alexander Grothendieck

Kombinatoristák

a program gyors legyen
teljesítmény
régi karbantartása
hacker
tartalom

Andrew Wiles

Formális nyelv

név ::= Mari | Jenő | Áron | Juli

alany ::= *név* | bicikli

tárgy ::= *alany*t

minőségjelző ::= Kis | Nagy | Szép

mennyiségjelző ::= három | egy | *mennyiségjelző* meg *mennyiségjelző*

állítmány ::= visz | kedvel

mondat ::= *minőségjelző alany mennyiségjelző tárgy állítmány*.

Mondat -e?

1. Szép Juli három biciklit kedvel.
2. Szép Juli három Áront kedvel.
3. Szép Juli három meg három Áront kedvel.
4. Szép Juli kedvel három biciklit.
5. Három Juli szép biciklit kedvel.

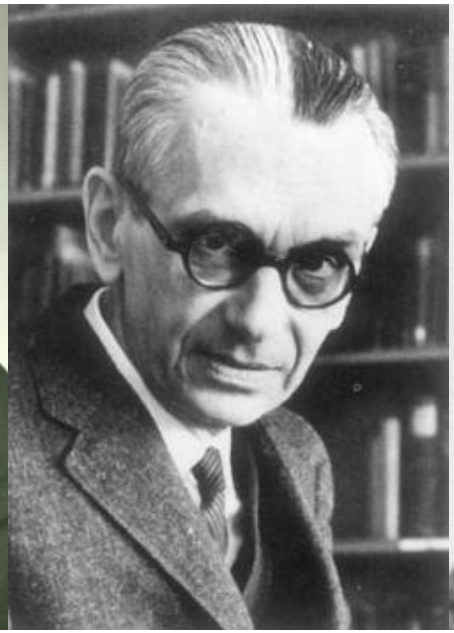
Továbbá: Szép Juli három meg (három meg három) Áront kedvel.

$\neq$ Szép Juli (három meg három) meg három Áront kedvel.

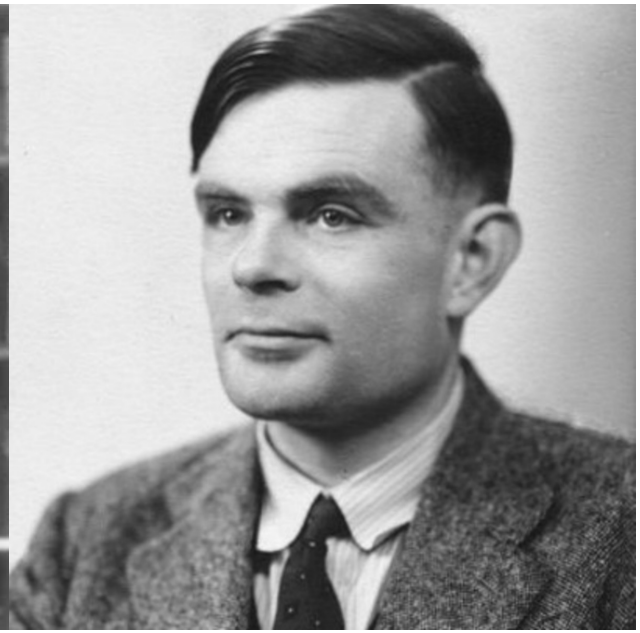
Lambda kalkulus (i)

Mi az, hogy program?

Alonzo Church



Kurt Gödel



Alan Turing

$$\begin{aligned} \textit{változó} &::= x \mid y \mid z \mid \textit{változó}' \\ \textit{program} &::= \textit{változó} \mid \lambda \textit{változó}.\textit{program} \mid \textit{program}\ \textit{program} \end{aligned}$$

Lambda kalkulus (ii)

Szintaxis rövidebben:

$$t ::= x \mid \lambda x.t \mid t\ t'$$

Példák:

$$\lambda x.x + x$$

$$(\lambda x.x + x)\, 3$$

Program végrehajtása:

$$(\lambda x.x + x)\, 3 \rightsquigarrow 3 + 3$$

Program több bemenettel:

$$\lambda x.(\lambda y.(2 * x + y))$$

Végrehajtás:

$$((\lambda x.(\lambda y.2 * x + y))\, 3)\, 4 \rightsquigarrow 2 * 3 + 4$$

Programok végrehajtása

Szintaxis:

$$t ::= x \mid \lambda x.t \mid t\ t'$$

Végrehajtás általános esetben, bármely x, t, t'-re:

$$(\lambda x.t)\ t' \rightsquigarrow t[x \mapsto t']$$

Példák:

$$(\lambda x.x + x)\ 3 \rightsquigarrow (x + x)[x \mapsto 3] = 3 + 3$$

$$\begin{aligned}
& ((\lambda x.(\lambda y.2 * x + y))\ 3)\ 4 \\
\rightsquigarrow\ & ((\lambda y.2 * x + y)[x \mapsto 3])\ 4 \\
=\ & (\lambda y.2 * 3 + y)\ 4 \\
\rightsquigarrow\ & (2 * 3 + y)[y \mapsto 4] \\
=\ & 2 * 3 + 4
\end{aligned}$$

Számok (i)

$$t ::= x \mid \lambda x.t \mid t\ t' \qquad\qquad (\lambda x.t)\ t' \rightsquigarrow t[x \mapsto t']$$

$$\begin{aligned} 0 &:= \lambda x.(\lambda y.x) \\ 1 &:= \lambda x.(\lambda y.(y\ x)) \\ 2 &:= \lambda x.(\lambda y.(y\ (y\ x))) \\ 3 &:= \lambda x.(\lambda y.(y\ (y\ (y\ x)))) \end{aligned}$$

$$\mathit{pluszegy} := \lambda z.(\lambda x.(\lambda y.y\ ((z\ x)\ y)))$$

Számok (ii)

$$t ::= x \mid \lambda x.t \mid t\ t' \qquad\qquad (\lambda x.t)\ t' \rightsquigarrow t[x \mapsto t']$$

$$2 = \lambda x.(\lambda y.(y\ (y\ x)))$$

$$\mathit{pluszegy} = \lambda z.(\lambda x.(\lambda y.y\ ((z\ x)\ y)))$$

Például:

$$\begin{aligned}
& \mathit{pluszegy}\ 2 \\
= \ & (\lambda z.(\lambda x.(\lambda y.y\ ((z\ x)\ y))))\ (\lambda x.(\lambda y.(y\ (y\ x)))) \\
\rightsquigarrow \ & (\lambda x.(\lambda y.y\ ((z\ x)\ y)))[z \mapsto (\lambda x.(\lambda y.(y\ (y\ x))))] \\
= \ & \lambda x.(\lambda y.y\ (((\lambda x.(\lambda y.(y\ (y\ x))))\ x)\ y)) \\
\rightsquigarrow \ & \lambda x.(\lambda y.y\ ((\lambda y.(y\ (y\ x)))[x \mapsto x]\ y)) \\
= \ & \lambda x.(\lambda y.y\ ((\lambda y.(y\ (y\ x)))\ y)) \\
\rightsquigarrow \ & \lambda x.(\lambda y.y\ ((y\ (y\ x))[y \mapsto y])) \\
= \ & \lambda x.(\lambda y.y\ (y\ (y\ x))) \\
= \ & 3
\end{aligned}$$

Házi feladat: összeadás, szorzás.

Típusok

Számokkal mindent lehet reprezentálni, például szövegeket, képeket, videókat stb.

Tfh. van egy nagy programunk, $\lambda x.\mathit{convert}$ alakú, bemenet: PNG, kimenet: JPG.
Van egy *rajz* programunk, ami egy PNG kép.
Ekkor $(\lambda x.\mathit{convert})\,\mathit{rajz}$ egy JPG típusú kép lesz.

Tfh. van egy *konyv* programunk, ami egy könyvnek a szövege.
Mi lesz, ha végrehajtjuk a $(\lambda x.\mathit{convert})\,\mathit{konyv}$ programot?

Típusok: PNG, JPG, Szöveg, Szám, Szám $\Rightarrow$ Szám, PNG $\Rightarrow$ JPG stb.

Típusrendszer (i)

$$\frac{\text{Ha } x : A, \text{ akkor } t : B}{\lambda x.t : A \Rightarrow B} \qquad \frac{t : A \Rightarrow B \qquad t' : A}{t\ t' : B}$$

$$\frac{\mathit{pluszegy} : \text{Szám} \Rightarrow \text{Szám} \qquad 2 : \text{Szám}}{\mathit{pluszegy}\ 2 : \text{Szám}}$$

$$\lambda x.\mathit{convert} : \text{PNG} \Rightarrow \text{JPG} \qquad \mathit{konyv} : \text{Szöveg}$$

Típusrendszer (ii)

$$\frac{\mathit{pluszegy} : \text{Szám} \Rightarrow \text{Szám} \qquad 2 : \text{Szám}}{\mathit{pluszegy}\ 2 : \text{Szám}}$$

Honnan tudjuk, hogy $2 : \text{Szám}$?

Kiegészítjük a típusrendszer szabályait:

$$\frac{}{0 : \text{Szám}} \qquad \frac{}{\mathit{pluszegy} : \text{Szám}}$$

$$\frac{\text{Ha } x : A, \text{ akkor } t : B}{\lambda x.t : A \Rightarrow B} \qquad \frac{t : A \Rightarrow B \qquad t' : A}{t\ t' : B}$$

A szabályok használatával:

$$\frac{\dfrac{}{\mathit{pluszegy} : \text{Szám} \Rightarrow \text{Szám}} \qquad \dfrac{\dfrac{}{\mathit{pluszegy} : \text{Szám} \Rightarrow \text{Szám}} \qquad \dfrac{}{0 : \text{Szám}}}{(\mathit{pluszegy}\ 0) : \text{Szám}}}{\mathit{pluszegy}\ (\mathit{pluszegy}\ 0) : \text{Szám}}$$

Matematika nyelve

Állítások, például: $(x + 1 \equiv 3) \Rightarrow (x \equiv 2)$

$(x \equiv 3 \wedge x \equiv 2) \Rightarrow \bot$

Logikai szabályok (egy része):

$$\frac{}{\top} \qquad \frac{\bot}{A} \qquad \frac{A \quad B}{A \wedge B} \qquad \frac{A \wedge B}{A} \qquad \frac{A \wedge B}{B} \qquad \frac{\begin{array}{c}A\\ \vdots \\ B\end{array}}{A \Rightarrow B} \qquad \frac{A \Rightarrow B \quad A}{B}$$

$$\frac{}{a \equiv a} \qquad \frac{a \equiv b \qquad b \equiv c}{a \equiv c} \qquad \frac{a \equiv b}{b \equiv a} \qquad \frac{a + 1 \equiv b + 1}{a \equiv b} \qquad \frac{a + 1 \equiv a}{\bot} \qquad \ldots$$

Bizonyítások:

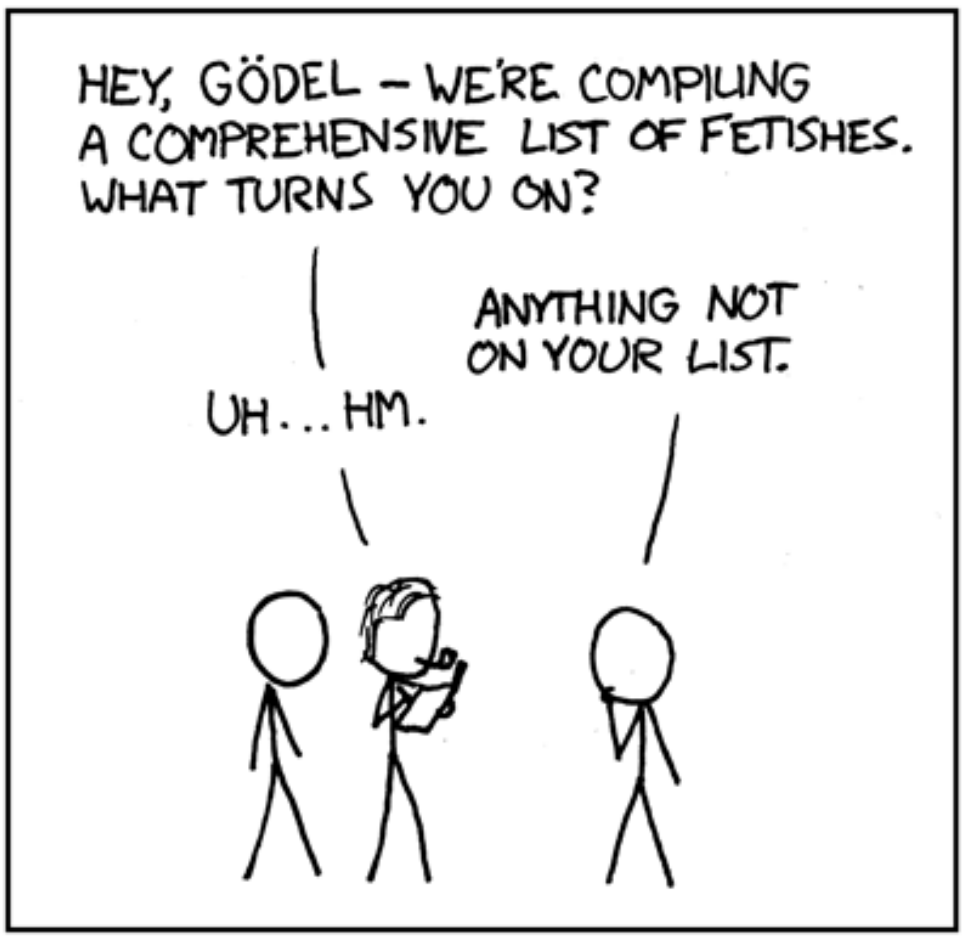
$$\frac{\dfrac{x + 1 \equiv 3}{x \equiv 2}}{(x + 1 \equiv 3) \Rightarrow (x \equiv 2)}$$

$$\frac{\dfrac{\dfrac{\dfrac{x \equiv 3 \wedge x \equiv 2}{x \equiv 3}}{3 \equiv x} \qquad \dfrac{x \equiv 3 \wedge x \equiv 2}{x \equiv 2}}{3 \equiv 2}}{(x \equiv 3 \wedge x \equiv 2) \Rightarrow \bot}$$

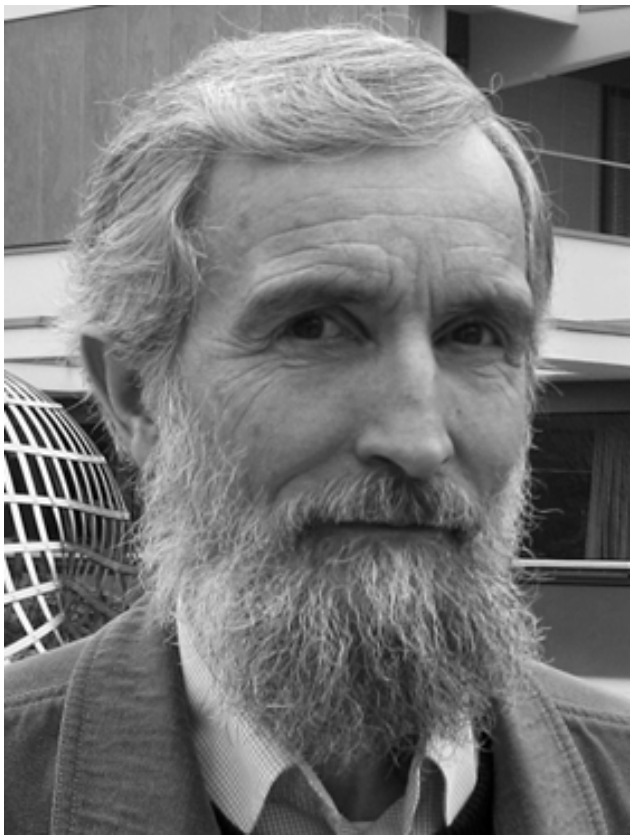
Kitérő: fel lehet -e írni az összes szabályt?

AUTHOR KATHARINE GATES RECENTLY ATTEMPTED TO MAKE A CHART OF ALL SEXUAL FETISHES.

LITTLE DID SHE KNOW THAT RUSSELL AND WHITEHEAD HAD ALREADY FAILED AT THIS SAME TASK.



Típuselmélet (type theory) vs. halmazelmélet



Per Martin-Löf filozófus, ornitológus.

A matematikai objektumok elválaszthatatlanok a típusuktól.
A bizonyítások konstruktívak.
A bizonyítások számítógéppel ellenőrizhetők.

Típusok

- $\mathbb{N}$: természetes számok
- $\Pi(x : A).B$: függvény, logikai következtetés, „minden” kvantor, általános szorzat
- $\Sigma(x : A).B$: rendezett pár, logikai és, „létezik” kvantor, általános összeg
- $a \equiv_A b$: egyenlőség

Például:

$$\lambda x.\lambda y.ap\ pred\ y \quad : \quad \Pi(x : \mathbb{N}).\Pi(y : x + 1 \equiv_{\mathbb{N}} 3).x \equiv_{\mathbb{N}} 2$$

$$\begin{aligned} \text{matematika} &= \text{programozás} \\ \text{állítás} &= \text{típus} \\ \text{bizonyítás} &= \text{program} \end{aligned}$$

Típuselméletre épülő progamozási nyelvek/tételbizonyító rendszerek: Agda, Coq (négy szín), Idris, Lean stb.

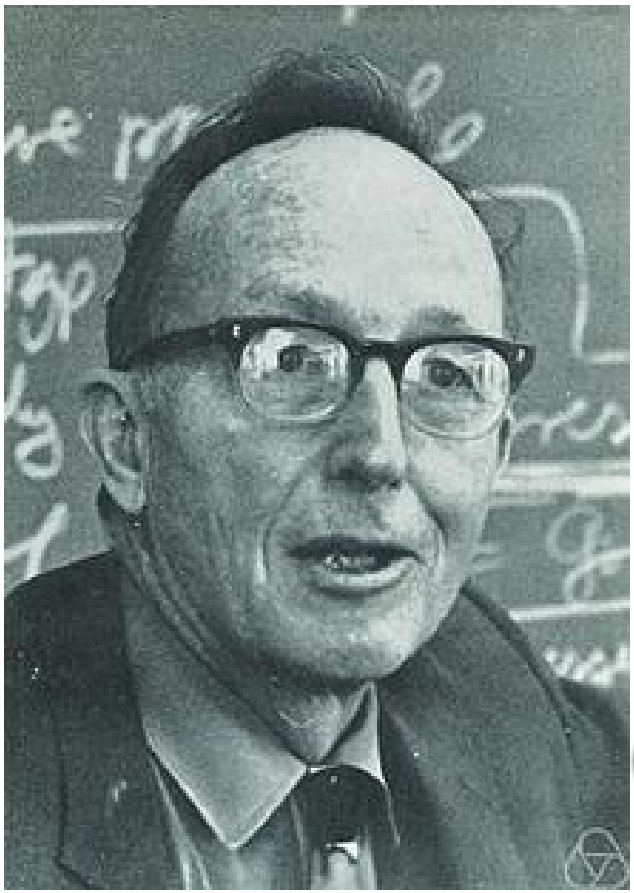
Egyenlőség típus

Kérdés: ha $t : a \equiv_A b$ és $t' : a \equiv_A b$, akkor vajon van -e olyan p, hogy $p : t \equiv_{a \equiv_A b} t'$?
Martin-Löf egységes sémát adott meg minden típusra, hogy milyen programokat lehet írni.
Például $\Pi(x : A).B$-re:

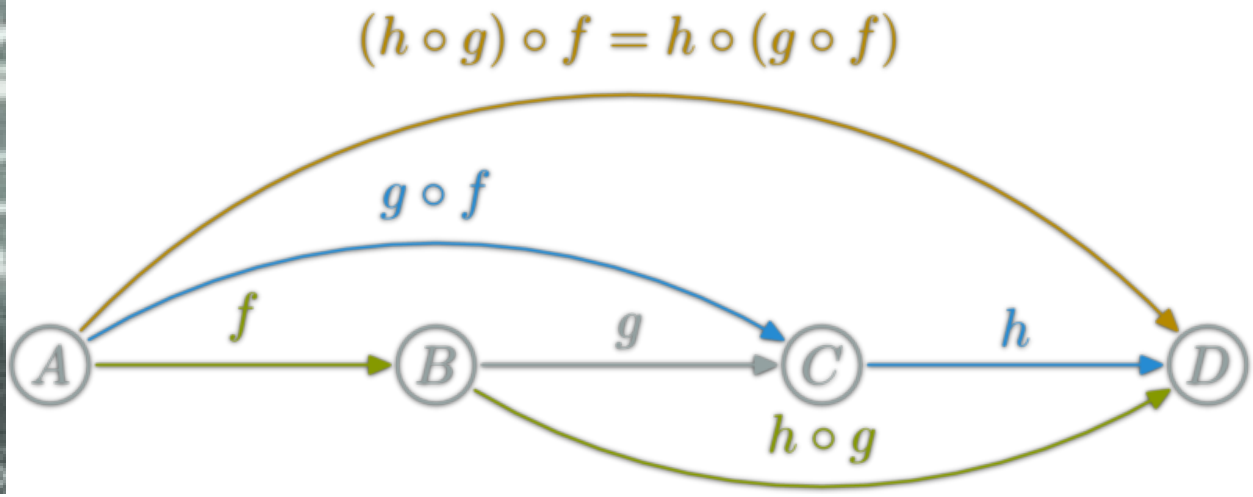
- típus formálás: ha A egy típus, és ha B típus, feltéve, hogy $x : A$, akkor $\Pi(x : A).B$ típus
- konstrukció: ha $t : B$, feltéve, hogy $x : A$, akkor $\lambda x.t : \Pi(x : A).B$
- elimináció: ha $t : \Pi(x : A).B$ és $u : A$, akkor $t\,u : B[x \mapsto u]$
- komputáció: $(\lambda x.t)\,u = t[x \mapsto u]$
- egyediség: $\lambda x.(t\,x) = t$

A hasonló, egyenlőségre vonatkozó sémából nem következik a fenti kérdésre igen válasz.

Kategóriaelmélet

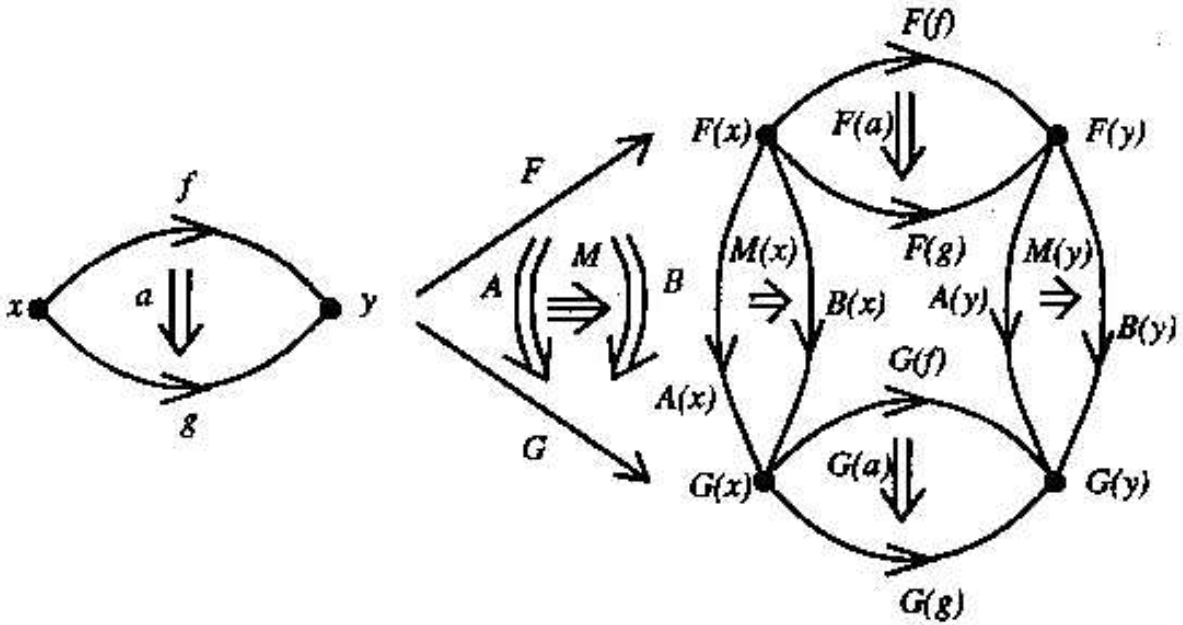


Saunders Mac Lane matematikus



Homotópia-elmélet

Vladimir Voevodsky matematikus



Mikor egyenlő két típus?

Ha van egy $f : A \Rightarrow B$ és egy $g : B \Rightarrow A$, melyekre $f \circ g = id_B$ és $g \circ f = id_A$.
Bool: kételemű típus, két konstruktorral: true : Bool, false: Bool

$$id \quad : Bool \Rightarrow Bool$$
$$id \;\; true = true$$
$$id \;\; false = false$$

$$not : Bool \Rightarrow Bool$$
$$not\, true = false$$
$$not\, false = true$$

Saját kutatás

Típuselmélet leírása a típuselméletben

Homotópia-típuselméletnek szép szintaxist készítünk

Matematikusokat meggyőzni, hogy formalizáljanak

Programozókat meggyőzni, hogy bizonyítottan helyes progamokat írjanak