# A legegyszerűbb programozási nyelv

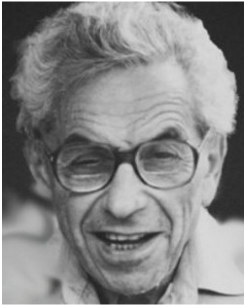
Kaposi Ambrus

Eötvös Loránd Tudományegyetem
Informatikai Kar

Kutatók Éjszakája
2019. szeptember 27.

## Kombinatoristák

a program gyors legyen
teljesítmény
régi karbantartása
hacker
tartalom
C



Erdős Pál

## Logisták

a program helyesen működjön
modularitás, absztrakció
új rendszer
nyelvész
forma
Haskell

Alexander Grothendieck

## Formális nyelv (i)

| | | |
|---|---|---|
| *név* | ::= | Mari \| Jenő \| Áron \| Juli |
| *alany* | ::= | *név* \| bicikli |
| *tárgy* | ::= | *alany*t |
| *minőségjelző* | ::= | Nagy \| Kék \| Szép |
| *mennyiségjelző* | ::= | egy \| három |
| *állítmány* | ::= | teker \| kedvel |
| *mondat* | ::= | *minőségjelző alany mennyiségjelző tárgy állítmány*. |

Mondat -e?

- Szép Juli egy biciklit teker.
- Szép Juli három Áront kedvel.
- Szép Juli kedvel egy Áront.
- Kék Juli egy Áront teker.
- Három Juli szép biciklit kedvel.

Hány mondat van?

# Formális nyelv (ii)

| | | |
|---|---|---|
| *név* | ::= | Mari \| Jenő \| Áron \| Juli |
| *alany* | ::= | *név* \| bicikli |
| *tárgy* | ::= | *alany*t |
| *minőségjelző* | ::= | Nagy \| Kék \| Szép |
| *mennyiségjelző* | ::= | egy \| három \| *mennyiségjelző* meg *mennyiségjelző* |
| *állítmány* | ::= | teker \| kedvel |
| *mondat* | ::= | *minőségjelző alany mennyiségjelző tárgy állítmány*. |

Hány mondat van?
Mondatok:

- Szép Juli egy meg három meg három Áront kedvel.
- Szép Juli egy meg (három meg három) Áront kedvel.
- Szép Juli (egy meg három) meg három Áront kedvel.

# Formális nyelv számokra

$$\textit{szám} ::= 0 \mid 1 \mid \textit{szám}\ \text{meg}\ \textit{szám}$$

- (1 meg 1) meg 1
- 1 meg (1 meg 1)
- ((1 meg 1) meg 1) meg 1
- (1 meg (1 meg 1)) meg 1
- 1 meg ((1 meg 1) meg 1)
- 1 meg (1 meg (1 meg 1))

Rajz!

# Formális nyelv számokra, redundancia nélkül (i)

$$\begin{aligned} \mathit{szám} ::= 0 \mid 1 \mid \mathit{szám}\ \text{meg}\ \mathit{szám} \mid \\ x\ \text{meg}\ (y\ \text{meg}\ z) = (x\ \text{meg}\ y)\ \text{meg}\ z \end{aligned}$$

$1\ \text{meg}\ (1\ \text{meg}\ 1) = (1\ \text{meg}\ 1)\ \text{meg}\ 1$

$x\ \text{meg}\ (y\ \text{meg}\ z) = (x\ \text{meg}\ y)\ \text{meg}\ z$

$$\begin{aligned}
1\ \text{meg}\ (1\ \text{meg}\ (1\ \text{meg}\ 1)) &= (1\ \text{meg}\ 1)\ \text{meg}\ (1\ \text{meg}\ 1) = ((1\ \text{meg}\ 1)\ \text{meg}\ 1)\ \text{meg}\ 1 \\
x\ \text{meg}\ (y\ \text{meg}\ z\phantom{\ \text{meg}\ 1)}) &= (x\ \text{meg}\ y)\ \text{meg}\ z \\
&\phantom{=(}x\phantom{\ \text{meg}\ 1)}\ \text{meg}\ (y\ \text{meg}\ z) = (x\phantom{(1\ \text{meg}\ 1)\ }\ \text{meg}\ y)\ \text{meg}\ z
\end{aligned}$$

Most már minden szám átasszociálható bármely változatára.
Biztos, hogy nincs redundancia? $0\ \text{meg}\ 1 \neq 1$

# Formális nyelv számokra, redundancia nélkül (ii)

$$
\begin{aligned}
\mathit{szám} ::= {} & 0 \,|\, 1 \,|\, \mathit{szám} \text{ meg } \mathit{szám} \,| \\
& x \text{ meg } (y \text{ meg } z) = (x \text{ meg } y) \text{ meg } z \,| \\
& 0 \text{ meg } x = x \,| \\
& x \text{ meg } 0 = x
\end{aligned}
$$

Most már minden természetes számhoz csak egyetlen *szám* tartozik:

- A nullák eltüntethetők. (Kivéve mikor?)
- Ha már nincsenek nullák, akkor átasszociálunk.

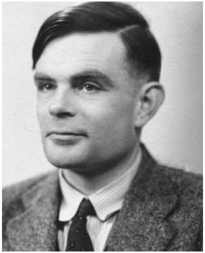
# Formális nyelv számokra, redundancia nélkül (iii)

Meg lehet-e a számokat adni egyenlőségek használata nélkül?

$$\mathit{szám} ::= 0 \mid \mathrm{M}\ \mathit{szám}$$

- $0 := 0$
- $1 := \mathrm{M}\ 0$
- $2 := \mathrm{M}\ (\mathrm{M}\ 0)$
- $3 := \mathrm{M}\ (\mathrm{M}\ (\mathrm{M}\ 0))$
- $\ldots$

# Mi az, hogy program?

|  |  |  |  |
|---|---|---|---|
| Moses Schönfinkel | Alonzo Church | Kurt Gödel | Alan Turing |
| kombinátor logika | lambda kalkulus | $\mu$-rekurzív függvények | Turing-gép |

# Formális nyelv programokra

$$P ::= \mathsf{I} \,|\, \mathsf{K} \,|\, \mathsf{S} \,|\, PP \,|\, \mathsf{I}x = x \,|\, \mathsf{K}xy = x \,|\, \mathsf{S}xyz = xz(yz)$$

Konvenció: $xyz := (xy)z$.

Soroljuk fel a programokat!

- I, K, S.
- II, IK, IS, KI, KK, KS, SI, SK, SS
- III, IIK, IIS, IKI, IKK, IKS, ISI, ISK,
  KII, KIK, KIS, KKI, KKK, KKS, KSI, KSK,
  SII, SIK, SIS, SKI, SKK, SKS, SSI, SSK
- I(II), I(IK), I(IS), I(KI), I(KK), I(KS), I(SI), I(SK), I(SS), K(II),
  K(IK), K(IS), K(KI), K(KK), K(KS), K(SI), K(SK), K(SS), S(II),
  S(IK), S(IS), S(KI), S(KK), S(KS), S(SI), S(SK), S(SS)
- ...

# Formális nyelv programokra

$$P ::= \mathsf{I} \,|\, \mathsf{K} \,|\, \mathsf{S} \,|\, PP \,|\, \mathsf{I}x = x \,|\, \mathsf{K}xy = x \,|\, \mathsf{S}xyz = xz(yz)$$

Konvenció: $xyz := (xy)z$.

Soroljuk fel a programokat!

- I, K, S.
- KI, KK, KS, SI, SK, SS
- SII, SIK, SIS, SKI, SKK, SKS, SSI, SSK
- K(KI), K(KK), K(KS), K(SI), K(SK), K(SS), S(KI), S(KK), S(KS), S(SI), S(SK), S(SS),
- ...

# Formális nyelv programokra

$$P ::= \mathsf{I} \,|\, \mathsf{K} \,|\, \mathsf{S} \,|\, PP \,|\, \mathsf{I}x = x \,|\, \mathsf{K}xy = x \,|\, \mathsf{S}xyz = xz(yz)$$

Konvenció: $xyz := (xy)z$.

Milyen programokat tudunk írni?

- Utolsó: $\mathsf{U}xy = y$
  Definíció: $\mathsf{U} := \mathsf{KI}$
  Teszt: $\mathsf{U}xy = \mathsf{KI}xy = \mathsf{I}y = y$
- Center: $\mathsf{C}xyz = y$
  Definíció: $\mathsf{C} := \mathsf{KK}$
  Teszt: $\mathsf{C}xyz = \mathsf{KK}xyz = \mathsf{K}yz = y$
- Duplázás: $\mathsf{D}x = xx$
  Visszafele definíció: $xx = \mathsf{I}x(\mathsf{I}x) = \mathsf{SII}x$, szóval $\mathsf{D} := \mathsf{SII}$

# Formális nyelv programokra

$$P ::= \mathsf{I} \,|\, \mathsf{K} \,|\, \mathsf{S} \,|\, PP \,|\, \mathsf{I}x = x \,|\, \mathsf{K}xy = x \,|\, \mathsf{S}xyz = xz(yz)$$

Eddigi programjaink tudása:

$$\mathsf{U}xy = y \qquad \mathsf{C}xyz = y \qquad \mathsf{D}x = xx$$

Néhány bonyolultabb program:

- Asszociálás: $\mathsf{A}fgx = f(gx)$
  Definíció: $\mathsf{A} := \mathsf{S}(\mathsf{KS})\mathsf{K}$
  Teszt:
  $\mathsf{A}fgx = \mathsf{S}(\mathsf{KS})\mathsf{K}fgx = \mathsf{KS}f(\mathsf{K}f)gx = \mathsf{S}(\mathsf{K}f)gx = \mathsf{K}fx(gx) = f(gx)$
- Fordít: $\mathsf{F}xy = yx$
  Definíció: $\mathsf{F} := \mathsf{S}(\mathsf{K}(\mathsf{SI}))(\mathsf{S}(\mathsf{KK})\mathsf{I})$
  Ellenőrizni!
- Végére ugrik: $\mathsf{V}xyz = xzy$
  Definíció: $\mathsf{G} := \mathsf{S}(\mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{KS})\mathsf{K}))\mathsf{S})(\mathsf{KK})$
  Ellenőrizni!

# Formális nyelv programokra

$$P ::= \mathsf{I} \,|\, \mathsf{K} \,|\, \mathsf{S} \,|\, PP \,|\, \mathsf{I}x = x \,|\, \mathsf{K}xy = x \,|\, \mathsf{S}xyz = xz(yz)$$

Eddigi programjaink tudása:

$\mathsf{U}xy = y \quad \mathsf{C}xyz = y \quad \mathsf{D}x = xx \quad \mathsf{A}fgx = f(gx) \quad \mathsf{F}xy = yx \quad \mathsf{V}xyz = xzy$

Számok Church-kódolása: $0fx = x,\ 1fx = fx,\ 2fx = f(fx),\ 3fx = f(f(fx))$

Implementáljuk! $0 := \mathsf{U}$

Meg egy: $\mathsf{M}afx = f(afx)$

$f(afx) = \mathsf{A}f(af)x = \mathsf{SA}afx$

$\mathsf{M} := \mathsf{SA}$

Összeadás: $\ddot{\mathsf{O}}abfx = af(bfx)$

$\ddot{\mathsf{O}} :=$ házi feladat