A legegyszerűbb programozási nyelv

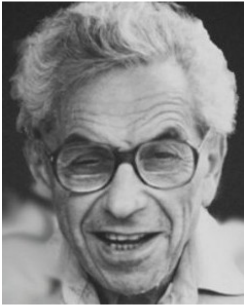
Kaposi Ambrus

Eötvös Loránd Tudományegyetem
Informatikai Kar

Kutatók Éjszakája
2021. szeptember 24.

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	::= Mózes | Alonso | Alan
alany	::= *név* | bicikli
tárgy	::= *alany*t
minőségjelző	::= Okos | Kék
mennyiségjelző	::= egy | három
állítmány	::= teker | fegyelmez
mondat	::= *minőségjelző alany mennyiségjelző tárgy állítmány*.

Mondat -e?

- Okos Mózes egy biciklit teker. IGEN
- Okos Mózes három Alant fegyelmez. IGEN
- Okos Mózes fegyelmez egy Alant. NEM
- Kék Mózes egy Alonsot teker. IGEN
- Három Alonso kék biciklit fegyelmez. NEM

Hány mondat van? $2 * (3 + 1) * 2 * 4 * 2 = 8 * 8 * 2 = 128$

Formális nyelv (ii)

név	::=	Mózes | Alonso | Alan
alany	::=	*név* | bicikli
tárgy	::=	*alany*t
minőségjelző	::=	Okos | Kék
mennyiségjelző	::=	egy | három | *mennyiségjelző* meg *mennyiségjelző*
állítmány	::=	teker | fegyelmez
mondat	::=	*minőségjelző alany mennyiségjelző tárgy állítmány*.

Hány mondat van?

- Okos Mózes egy Alant fegyelmez.
- Okos Mózes egy meg egy Alant fegyelmez.
- Okos Mózes egy meg egy meg egy Alant fegyelmez.
- Okos Mózes egy meg egy meg egy meg egy Alant fegyelmez.
- ...

Végtelen sok!

Csak a forma számít (i)

név	::=	Imre | Kati | Marci
alany	::=	*név* | trambulin
tárgy	::=	*alany*t
minőségjelző	::=	Nagy | Ordítós
mennyiségjelző	::=	sok | kevés | *mennyiségjelző* mínusz *mennyiségjelző*
állítmány	::=	okít | előreenged
mondat	::=	*minőségjelző alany mennyiségjelző tárgy állítmány*.

- Nagy Imre sok Marcit előreenged.
- Nagy Imre sok mínusz sok Marcit előreenged.
- Nagy Imre sok mínusz sok mínusz sok Marcit előreenged.
- Nagy Imre sok mínusz sok mínusz sok mínusz sok Marcit előreenged.
- ...

Csak a forma számít (ii)

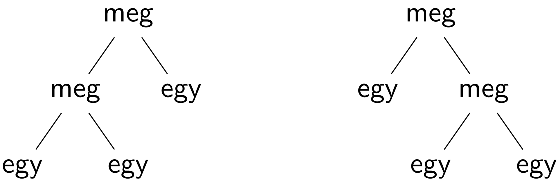
$$
\begin{array}{ll}
A & ::= \mathrm{a} \mid \mathrm{b} \mid \mathrm{c} \\
B & ::= A \mid \mathrm{d} \\
C & ::= B\ \mathrm{e} \\
D & ::= \mathrm{f} \mid \mathrm{g} \\
E & ::= \mathrm{h} \mid \mathrm{i} \mid E\ \mathrm{j}\ E \\
F & ::= \mathrm{k} \mid \mathrm{l} \\
G & ::= \mathrm{m}\ D \quad B \quad E \quad C \quad F
\end{array}
$$

- m f a h c e l
- m f a h j h c e l
- m f a h j h j h c e l
- m f a h j h j h j h c e l
- ...

Formális nyelv számokra (i)

$$\textit{szám} ::= \text{nulla} \mid \text{egy} \mid \textit{szám} \text{ meg } \textit{szám}$$

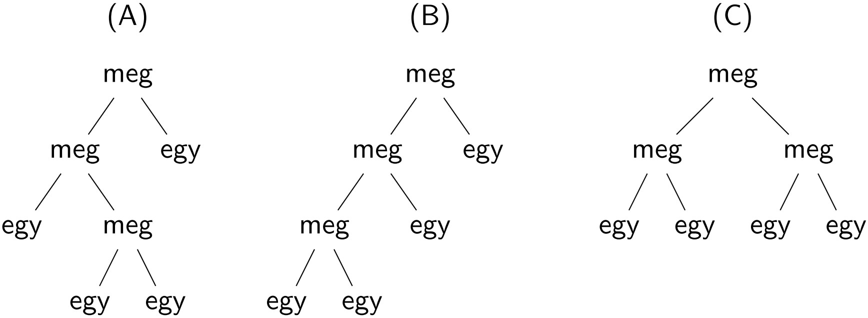
- nulla, egy, egy meg egy, egy meg egy meg egy, . . .
- (egy meg egy) meg egy $\neq$ egy meg (egy meg egy)



- Okos Mózes (egy meg egy) meg egy Alant fegyelmez $\neq$ Okos Mózes egy meg (egy meg egy) Alant fegyelmez

Formális nyelv számokra (ii)

szám ::= nulla | egy | *szám* meg *szám*



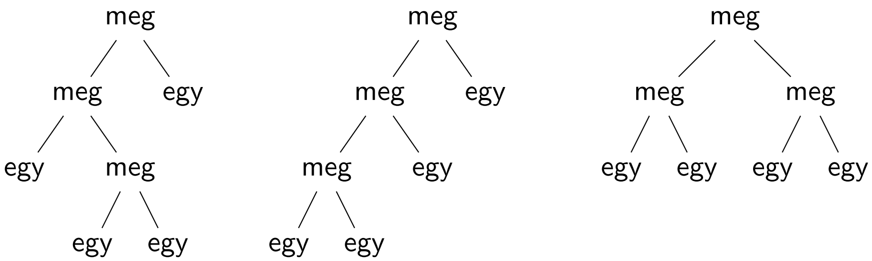
Melyik fához megyik szöveg tartozik?

1. ( ) (egy meg egy) meg (egy meg egy)
2. ( ) ((egy meg egy) meg egy) meg egy
3. ( ) (egy meg (egy meg egy)) meg egy

Formális nyelv számokra (ii)

szám ::= nulla | egy | *szám* meg *szám*

(A) (B) (C)



Melyik fához megyik szöveg tartozik?

1. (C) (egy meg egy) meg (egy meg egy)
2. (B) ((egy meg egy) meg egy) meg egy
3. (A) (egy meg (egy meg egy)) meg egy

Formális nyelv számokra, redundancia nélkül (i)

$$\textit{szám} ::= \text{nulla} \mid \text{egy} \mid \textit{szám} \text{ meg } \textit{szám} \mid$$

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

egy meg (egy meg egy) = (egy meg egy) meg egy

x meg (y meg z) = (x meg y) meg z

egy meg (egy meg (egy meg egy)) = (egy meg egy) meg (egy meg egy)

x meg (y meg z) = (x meg y) meg z

(egy meg egy) meg (egy meg egy) = ((egy meg egy) meg egy) meg egy

x meg (y meg z) = (x meg y) meg z

Formális nyelv számokra, redundancia nélkül (ii)

$$\textit{szám} ::= \text{nulla} \mid \text{egy} \mid \textit{szám} \text{ meg } \textit{szám} \mid$$

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

Most már minden szám átasszociálható ilyen alakra:

$$((\ldots((\text{egy meg egy}) \text{ meg egy})\ldots) \text{ meg egy}) \text{ meg egy}$$

De még mindig van egy kis redundancia:

$$\text{egy} \neq \text{egy meg nulla} \neq \text{nulla meg egy} \neq \text{nulla meg nulla meg egy}$$

Az lenne a jó, ha a nullák hozzáadása nem változtatni a számon.

Formális nyelv számokra, redundancia nélkül (iii)

$$\begin{aligned} \textit{szám} ::= {} & \text{nulla} \mid \text{egy} \mid \textit{szám} \text{ meg } \textit{szám} \mid \\ & x \text{ meg } (y \text{ meg } z) = (x \text{ meg } y) \text{ meg } z \mid \\ & x \text{ meg nulla} = x \mid \\ & \text{nulla meg } x = x \end{aligned}$$

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

Az összes szám:

- nulla
- egy
- egy meg egy
- (egy meg egy) meg egy
- ...

Formális nyelv számokra, redundancia nélkül (iv)

$$
\begin{aligned}
\mathit{szám} ::= {} & \text{nulla} \mid \text{egy} \mid \mathit{szám} \text{ meg } \mathit{szám} \mid \\
& x \text{ meg } (y \text{ meg } z) = (x \text{ meg } y) \text{ meg } z \mid \\
& x \text{ meg nulla} = x \mid \\
& \text{nulla meg } x = x
\end{aligned}
$$

Meg lehet-e a számokat adni egyenlőségek és redundancia nélkül?

$$\mathit{szám} ::= \text{nulla} \mid \mathit{szám} \text{ megegy}$$

- nulla
- nulla megegy
- (nulla megegy) megegy
- ((nulla megegy) megegy) megegy
- ...

Ezek a fák hogy néznek ki?

Mi az, hogy program?

Moses Schönfinkel

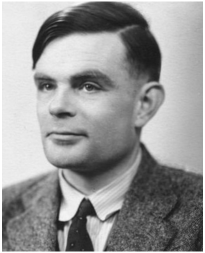
kombinátor logika

Alonzo Church

lambda kalkulus

Kurt Gödel

μ-rekurzív függvények



Alan Turing

Turing-gép

Moses Schönfinkel formális nyelve

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

Egyezményes jelölés: $xyz := (xy)z$.

Soroljuk fel a programokat!

- K, S
- KK, KS, SK, SS
- (KK)K, (KK)S, (KS)K, (KS)S, (SK)K, (SK)S, (SS)K, (SS)S
- K(KK), K(KS), K(SK), K(SS), S(KK), S(KS), S(SK), S(SS)
- ((KK)K)K, . . . , ((SS)S)S
- (K(KK))K, . . . , (S(SS))S
- (KK)(KK), . . . , (SS)(SS)
- K((KK)K), . . . , S((SS)S)
- K(K(KK)), . . . , S(S(SS))

Moses Schönfinkel formális nyelve

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

Egyezményes jelölés: $xyz := (xy)z$.

Soroljuk fel a programokat!

- K, S
- KK, KS, SK, SS
- SKK, SKS, SSK, SSS
- K(KK), K(KS), K(SK), K(SS), S(KK), S(KS), S(SK), S(SS)
-
- S(KK)K, . . . , S(SS)S
- (SK)(KK), . . . , (SS)(SS)
- K(SKK), . . . , S(SSS)
- K(K(KK)), . . . , S(S(SS))

Programozás

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

- Identitás: $\mathsf{I}x = x$
  Definíció: $\mathsf{I} := \mathsf{SKK}$
  Próbáljuk ki! $\mathsf{I}x = \mathsf{SKK}x = \mathsf{K}x(\mathsf{K}x) = x$
- Utolsó: $\mathsf{U}xy = y$
  Definíció: $\mathsf{U} := \mathsf{KI}$
  Teszt: $\mathsf{U}xy = \mathsf{KI}xy = \mathsf{I}y = y$
- Duplázás: $\mathsf{D}x = xx$
  Visszafele definíció: $xx = \mathsf{I}x(\mathsf{I}x) = \mathsf{SII}x$, szóval $\mathsf{D} := \mathsf{SII}$
- Balról asszociál: $\mathsf{B}xyz = x(yz)$
  Definíció: $\mathsf{B} := \mathsf{S(KS)K}$
  $\mathsf{B}xyz = \mathsf{S(KS)K}xyz = \mathsf{KS}x(\mathsf{K}x)yz = \mathsf{S}(\mathsf{K}x)yz = \mathsf{K}xz(yz) = x(yz)$
- Csere: $\mathsf{C}xyz = xzy$
  Definíció: $\mathsf{C} := \mathsf{S(S(KB)S)(KK)}$
- Fordít: $\mathsf{F}xy = yx$
  Definíció: $\mathsf{F} := \mathsf{S(K(SI))(S(KK)I)}$

Számok Schönfinkel nyelvén

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

Eddigi programjaink tudása (library/programkönyvtár interface-e):

$$\mathsf{I}x = x \quad \mathsf{U}xy = y \quad \mathsf{D}x = xx \quad \mathsf{B}xyz = x(yz) \quad \mathsf{C}xyz = xzy \quad \mathsf{F}xy = yx$$

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

Implementáljuk! $0 := \mathsf{U}$

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

Visszafele: $f(afx) = \mathsf{B}f(af)x = \mathsf{SB}afx$

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

Megvan az összes szám: $0 := \mathsf{U}, 1 := \mathsf{MU}, 2 := \mathsf{M(MU)}, 3 := \mathsf{M(M(MU))}$

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

$$af(bfx) = \mathsf{B}(af)(bf)x = \mathsf{BB}af(bf)x = \mathsf{S}(\mathsf{BB}a)bfx = \mathsf{BS}(\mathsf{BB})abfx$$

$$\ddot{\mathsf{O}} := \mathsf{BS}(\mathsf{BB})$$