

# A legegyszerűbb programozási nyelv

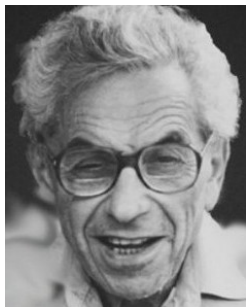
Kaposi Ambrus

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* ::= *alanyt*

*minőségjelző* ::= Nagy | Kék | Szép

*menyiségjelző* ::= 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* ::= *alanyt*

*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

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

- ▶ (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)

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

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

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

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

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

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

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

Most már minden szám átasszociálható bármely változatára.

Biztos, hogy nincs redundancia?  $0 \text{ meg } 1 \neq 1$

## Formális nyelv számokra, redundancia nélkül (ii)

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

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?

$szám ::= 0 \mid M \ szám$

- ▶  $0 := 0$
- ▶  $1 := M \ 0$
- ▶  $2 := M \ (M \ 0)$
- ▶  $3 := M \ (M \ (M \ 0))$
- ▶ ...



# Mi az, hogy program?



Moses Schönfinkel

kombinátor logika



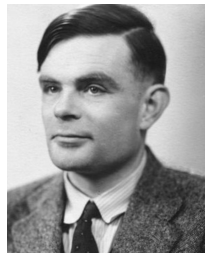
Alonzo Church

lambda kalkulus



Kurt Gödel

$\mu$ -rekurzív  
függvények



Alan Turing

Turing-gép

## Formális nyelv programokra

$$P ::= I \mid K \mid S \mid PP \mid Ix = x \mid Kxy = x \mid Sxyz = xz(yz)$$

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

$$P ::= I \mid K \mid S \mid PP \mid Ix = x \mid Kxy = x \mid Sxyz = xz(yz)$$

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

$$P ::= I \mid K \mid S \mid PP \mid Ix = x \mid Kxy = x \mid Sxyz = xz(yz)$$

Konvenció:  $xyz := (xy)z$ .

Milyen programokat tudunk írni?

► Utolsó:  $Uxy = y$

Definíció:  $U := KI$

Teszt:  $Uxy = KIxy = Iy = y$

► Center:  $Cxyz = y$

Definíció:  $C := KK$

Teszt:  $Cxyz = KKxyz = Kyz = y$

► Duplázás:  $Dx = xx$

Visszafele definíció:  $xx = Ix(Ix) = SIIx$ , szóval  $D := SII$

## Formális nyelv programokra

$$P ::= I \mid K \mid S \mid PP \mid Ix = x \mid Kxy = x \mid Sxyz = xz(yz)$$

Eddigi programjaink tudása:

$$Uxy = y \quad Cxyz = y \quad Dx = xx$$

Néhány bonyolultabb program:

- ▶ Asszociálás:  $Afgx = f(gx)$

Definíció:  $A := S(KS)K$

Teszt:

$$Afgx = S(KS)Kfgx = KSf(Kf)gx = S(Kf)gx = Kfx(gx) = f(gx)$$

- ▶ Fordít:  $Fxy = yx$

Definíció:  $F := S(K(SI))(S(KK)I)$

Ellenőrizni!

- ▶ Végére ugrik:  $Vxyz = xzy$

Definíció:  $G := S(S(K(S(KS)K))S)(KK)$

Ellenőrizni!

## Formális nyelv programokra

$$P ::= I \mid K \mid S \mid PP \mid Ix = x \mid Kxy = x \mid Sxyz = xz(yz)$$

Eddigi programjaink tudása:

$$U_{xy} = y \quad C_{xyz} = y \quad D_x = xx \quad Afx = f(gx) \quad F_{xy} = yx \quad V_{xyz} = xzy$$

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

Implementáljuk!  $0 := F$

Meg egy:  $Mafx = f(afx)$   
 $f(afx) = Af(af)x = SAafx$   
 $M := SA$

Összeadás:  $\ddot{O}abfx = af(bfx)$   
 $\ddot{O} := \text{házi feladat}$