

# Az esport kultúra nyelvének megalkotása: egy előtanulmány

## Construction of the Language of the Esports Culture: A Preliminary Study

Bátfai Norbert

`batfai.norbert@inf.unideb.hu`

Információ Technológia Tanszék, Debreceni Egyetem,  
Magyarország  
2019. szeptember 30.

### Kivonat

A Paszigráfia Rapszódia (Bátfai, 2019b), vagy röviden PaRa egy olyan mesterséges nyelv kialakítására törekvő kezdeményezés, mely lehetővé teszi a homunkulusz és a mesterséges homunkulusz közötti kommunikációt, ergo ez az esport kultúra nyelve (Bátfai, 2019a). Ebben az előtanulmányban megpróbálkozunk ennek a nyelvnek a megalkotásával.

### Abstract

The Pasigraphy Rhapsody (Bátfai, 2019b), or shortly PaRa (from it's Hungarian name Paszigráfia Rapszódia) is an initiative to develop an artificial language that is intended to allow communication between the Homunculus and the Artificial Homunculus, ergo, it is the language of the esport culture (Bátfai, 2019a). In this preliminary study, we have tried to construct this language.

## 1. Bevezetés

A PaRa vizualizációjának (Bátfai, 2019c) alapját az SMNIST képek (Bátfai et al., 2019) képezik, ahol a képeken lévő pöttyök számosságát kell meghatározni. Ám meg kell jegyezzük, hogy a PaRa esetében nem a számosság, hanem a pöttyök pontos helyzete a meghatározó.

## 2 Introduction

The intuitive base of visualization of PaRa (Bátfai, 2019c) comes from SMNIST images (Bátfai et al., 2019), where the numerosity of dots must be recognized in images. But it should be noticed that in the case of PaRa the numerosity does not matter but the exact position of the dots.

## 2.1. Első próbálkozások

NP Legyen adott egy természetes nyelvű  $s$  mondat.

FOL Próbálkozzunk meg  $s$  elsőrendű logikai formalizálásával, nevezzünk ezt  $F(s)$ -nek ami a Paszigráfia Rapszódia előnyelvének mondata lesz. Ez azt jelenti, hogy  $s$  jelentését meg kell próbálnunk kifejezni a PaRa előnyelvén. Itt arra rá kell mutatnunk, hogy ez a folyamat bem algoritmizálható. Ezt a lépést mindig a humán játékos hajtja végre.

TILE Majd rajzoljuk át  $F(s)$ -t a PaRa csempéző rácsba.

NUM Convert cells of the tiling grid into unique numbers.

SMNIST Convert these unique numbers into SMNIST codes.

NDIM Végül egy  $n$  dimenziós hiperkocka formájában vizualizáljuk az eredményt.

## 2.2. First Attempts

NP Let be given a natural language sentence  $s$ .

FOL Try to find a first order logic formalization for  $s$ . Let's called it  $F(s)$  that is a sentence of the proto-language of Pasigraphy Rhapsody. This means that the meaning of  $s$  must be tried to express in the proto-language of PaRa. It should be noticed that this process cannot be algorithmized. This step is always performed by the human player. But the following steps labelled by TILE, NUM, SMNIST and NDIM can be fully automated.

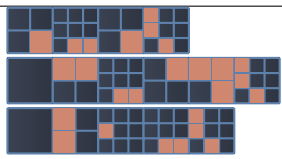
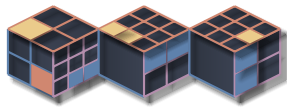
TILE Then draw  $F(s)$  in the form of PaRa tiling grid.

NUM Convert cells of the tiling grid into unique numbers.

SMNIST Convert these unique numbers into SMNIST codes.

NDIM Finally visualize the results as an  $n$ -dimensional hypercube.

### 2.1. Example.

English	PaRa 2D	PaRa 3D
Mice hate cats.		

### 2.2.1. The syntax of the proto-language of Pasigraphy Rhapsody

Here we use the notations of Albert Dragálin (Dragálin & Buzási, 1996) and the book (Russell & Norvig, 2010) together.

#### Types

$$\langle \text{Index} \rangle \models \mathbf{I} \langle \text{Index} \rangle \mid \mathbf{I} \quad (1)$$

$$\langle \text{Srt} \rangle \models \langle \text{Index} \rangle \quad (2)$$

### Variables

$$\langle \text{Variable} \rangle \models (\langle \text{Srt} \rangle . \langle \text{Index} \rangle) \quad (3)$$

### Constants

$$\langle \text{Cnst} \rangle \models (\text{C} . \langle \text{Srt} \rangle . \langle \text{Index} \rangle) \quad (4)$$

### Functions

$$\langle \text{Fn} \rangle \models (\text{F} . \langle \text{Index} \rangle \langle \text{ParamList} \rangle) \quad (5)$$

$$\langle \text{ParamList} \rangle \models \lambda \mid (\langle \text{ParamListBuilder} \rangle) \quad (6)$$

$$\langle \text{ParamListBuilder} \rangle \models \langle \text{Term} \rangle \mid \langle \text{Term} \rangle, \langle \text{ParamListBuilder} \rangle \quad (7)$$

### Predicates

$$\langle \text{Pr} \rangle \models (\text{P} . \langle \text{Index} \rangle \langle \text{ParamList} \rangle) \quad (8)$$

### Terms

$$\langle \text{Term} \rangle \models \langle \text{Variable} \rangle \mid \langle \text{Cnst} \rangle \mid \langle \text{Fn} \rangle \quad (9)$$

### Atomic Formulas

$$\langle \text{Atomic Formula} \rangle \models \langle \text{Pr} \rangle \quad (10)$$

### Formulas

$$\begin{aligned} \langle \text{Formula} \rangle &\models \langle \text{Atomic Formula} \rangle \mid \neg \langle \text{Formula} \rangle \mid \\ \langle \text{Formula} \rangle &\langle \text{Connective} \rangle \langle \text{Formula} \rangle \mid \end{aligned} \quad (12)$$

$$\langle \text{Quantifier} \rangle \langle \text{Variable} \rangle \langle \text{Formula} \rangle \quad (13)$$

### Logical Connectives

$$\langle \text{Connective} \rangle \models \wedge \mid \vee \mid \supset \quad (14)$$

### Quantifiers

$$\langle \text{Quantifier} \rangle \models \exists \mid \forall \quad (15)$$

**2.1. Példa.** Formalizáljuk a következő természetes nyelvi kijelentést!

Az egerek utálják a macskákat.

$$\forall \text{Állat}.x \forall \text{Állat}.y (\text{P.Egér}(x) \wedge \text{P.Macska}(y) \supset \text{P.Utál}(x, y))$$

Kicsit jobban megfeleltetve a grammatikának azt íránk, hogy

$$\forall(1.1) \forall(1.2) ((P.1(1.1)) \wedge (P.2(1.2)) \supset (P.3(1.1, 1.2)))$$

szóval mindent szépen számozgatunk, itt például minden változót és prédikátumot külön, a felbukkanásuk rendjében. Ha ez a mondat volt a "Helló, Világ!"<sup>1</sup> mondatunk, akkor most éppen ezeket a táblázatokat kell karban tartanunk:

Típus	Prédikátum
Állat=1	Egér=1
	Macska=2
	Utál=3

Méginkább közelítve a grammatikához pedig ezt íránk

$$\forall(I)\forall(II)((P.I(II)) \wedge (P.II(II)) \supset (P.II(I,II)))$$

ami még mindig elég pongyola, de pontosabbá már a grammmatikával együttesen történő pontosítással tehetnénk (például a zárójelezés tekintetében), de nem tesszük, mert a formalizálásoknál úgylis maradunk az első formánál, s számozni is egészen máshogy fogunk. Tehát a grammatikával mindössze annyit rögzítettünk, hogy elvben megszámlálhatóan végtelen sok típust, változót, konstant, prédikátumot és függvényt akarunk felhasználni.

### 2.2.2. The PaRa Tiling Language

A zárójelezést a behúzással váltjuk ki.

$\forall$	$x$	$\forall$	$y$		
	$E$	$x$	$\wedge$	$M$	$y$
	$\supset$	$U$	$x$	$y$	

#### 2.2. Példa.

Kicsit kevésbé leegyszerűsítve.

---

<sup>1</sup>Tehát az első formalizált mondatunk.

$\forall$	$x$	$\forall$	$y$		
		$E$			
			$x$		
		$\wedge$			
		$M$			
			$y$		
	$\supset$				
		$U$			
			$x$	$y$	

Persze ki kellene írunk a grammatika terminális ?karácsonyfáit? (például a típusokat és a sorszámot elválasztó pontot, vagy a predikátumokat bevezető P betűt stb.), de nem ezt fogjuk tenni, hanem bevezetünk egy egységes számozási rendszert.

### 2.2.3. The PaRa Numeration System

Terminális betű	PaRa számkód
$\exists$	1
$\forall$	2
$\neg$	3
$\wedge$	4
$\vee$	5
$\supset$	6

Maradt még megszámlálhatóan végtelen predikátum, függvény és típus nevünk, illetve minden megszámlálható típuson belül ugyancsak megszámlálható konstans és változó nevünk. Ezeket is könnyen számba vehetjük például Péter Rózsa könyvéből a transzfinit indukció bevezetése kapcsán említett, a természetes számokat megszámlálható sok diszjunk részhamazra bontását bemutató példájával, azaz

- A  $PaRa_{\text{text}} = \{7, 9, 11, 13, 15, 17, \dots\}$  halmaz, avagy a páratlan számokból kiszedve a terminális betűk kódjait, jelöli a már előállított PaRa szövegek indexeit. Minden előállított PaRa szöveget (például egy PaRa csempézési mátrixot, vagy a formalizált PaRa előnyelvi formulákat) megjegyezzünk egy listában, ennek indexét adja a páratlan szám eképpen:  $\frac{\text{páratlan szám}-1}{2} - 2$ .

- A  $PaRa_{\text{pred}} = \{10, 14, 18, 22, 26, \dots\}$  halmaz, avagy a kettőnek csak az első hatványával oszthatóak (megint csak kiszedve a terminálisok) jelölik a PaRa prédikátumokat.
- A  $PaRa_{\text{fn}} = \{12, 20, 28, 36, 44, \dots\}$  halmaz, avagy a kettőnek legfeljebb csak a második hatványával oszthatóak (ugyancsak kiszedve a terminálisok) jelölik a PaRa függvényeket.
- A  $PaRa_{\text{srt}} = \{8, 24, 40, 56, 72, \dots\}$  halmaz, avagy a kettőnek legfeljebb csak a harmadik hatványával oszthatóak jelölik a PaRa típusokat.
- A  $PaRa_{c_1} = \{16, 48, 80, 112, 144, \dots\}$  halmaz, avagy a kettőnek legfeljebb csak a negyedik hatványával oszthatóak jelölik a PaRa első típusának végtelen sok konstansát.
- A  $PaRa_{v_1} = \{32, 96, 160, 224, 288, \dots\}$  halmaz, avagy a kettőnek legfeljebb csak a ötödik hatványával oszthatóak jelölik a PaRa első típusának végtelen sok változóját.
- ...
- A  $PaRa_{c_n} = \{2^{2n+2}, 2^{2n+2} + 2^{2n+3}, 2^{2n+2} + 2 * 2^{2n+3}, 2^{2n+2} + 3 * 2^{2n+3}, \dots\}$  halmaz, avagy a kettőnek legfeljebb csak az  $2n+2$ . hatványával oszthatóak jelölik a PaRa  $n$ . típusának végtelen sok konstansát.
- A  $PaRa_{v_n} = \{2^{2n+3}, 2^{2n+3} + 2^{2n+4}, 2^{2n+3} + 2 * 2^{2n+4}, 2^{2n+3} + 3 * 2^{2n+4}, \dots\}$  halmaz, avagy a kettőnek legfeljebb csak az  $2n+3$ . hatványával oszthatóak jelölik a PaRa  $n$ . típusának végtelen sok változóját.
- ...

**2.3. Példa.**

2	32	2	96		
	10	32	4	14	96
	6	18	32	96	

2	32	2	96		
		10			
			32		
		4			
		14			
			96		
	6				
		18			
			32	96	

#### 2.2.4. SMNIST codes

```

n: 4 k: 1 index: 1 1: (0,0)
n: 4 k: 1 index: 2 1: (1,0)
n: 4 k: 1 index: 3 1: (0,1)
n: 4 k: 1 index: 4 1: (1,1)
*****
n: 4 k: 2 index: 5 1: (0,0) 2: (1,0)
n: 4 k: 2 index: 6 1: (0,0) 2: (0,1)
n: 4 k: 2 index: 7 1: (0,0) 2: (1,1)
n: 4 k: 2 index: 8 1: (1,0) 2: (0,1)
n: 4 k: 2 index: 9 1: (1,0) 2: (1,1)
n: 4 k: 2 index: 10 1: (0,1) 2: (1,1)
*****
n: 4 k: 3 index: 11 1: (0,0) 2: (1,0) 3: (0,1)
n: 4 k: 3 index: 12 1: (0,0) 2: (1,0) 3: (1,1)
n: 4 k: 3 index: 13 1: (0,0) 2: (0,1) 3: (1,1)
n: 4 k: 3 index: 14 1: (1,0) 2: (0,1) 3: (1,1)
*****
n: 9 k: 1 index: 15 1: (0,0)
n: 9 k: 1 index: 16 1: (1,0)
n: 9 k: 1 index: 17 1: (2,0)
n: 9 k: 1 index: 18 1: (0,1)
n: 9 k: 1 index: 19 1: (1,1)
n: 9 k: 1 index: 20 1: (2,1)
n: 9 k: 1 index: 21 1: (0,2)
n: 9 k: 1 index: 22 1: (1,2)

```

```

n: 9 k: 1 index: 23 1: (2,2)
*****
n: 9 k: 2 index: 24 1: (0,0) 2: (1,0)
n: 9 k: 2 index: 25 1: (0,0) 2: (2,0)
n: 9 k: 2 index: 26 1: (0,0) 2: (0,1)
n: 9 k: 2 index: 27 1: (0,0) 2: (1,1)
n: 9 k: 2 index: 28 1: (0,0) 2: (2,1)
n: 9 k: 2 index: 29 1: (0,0) 2: (0,2)
n: 9 k: 2 index: 30 1: (0,0) 2: (1,2)
n: 9 k: 2 index: 31 1: (0,0) 2: (2,2)
n: 9 k: 2 index: 32 1: (1,0) 2: (2,0)
n: 9 k: 2 index: 33 1: (1,0) 2: (0,1)
n: 9 k: 2 index: 34 1: (1,0) 2: (1,1)
n: 9 k: 2 index: 35 1: (1,0) 2: (2,1)
n: 9 k: 2 index: 36 1: (1,0) 2: (0,2)
n: 9 k: 2 index: 37 1: (1,0) 2: (1,2)
n: 9 k: 2 index: 38 1: (1,0) 2: (2,2)
n: 9 k: 2 index: 39 1: (2,0) 2: (0,1)
n: 9 k: 2 index: 40 1: (2,0) 2: (1,1)
n: 9 k: 2 index: 41 1: (2,0) 2: (2,1)
n: 9 k: 2 index: 42 1: (2,0) 2: (0,2)
n: 9 k: 2 index: 43 1: (2,0) 2: (1,2)
n: 9 k: 2 index: 44 1: (2,0) 2: (2,2)
n: 9 k: 2 index: 45 1: (0,1) 2: (1,1)
n: 9 k: 2 index: 46 1: (0,1) 2: (2,1)
n: 9 k: 2 index: 47 1: (0,1) 2: (0,2)
n: 9 k: 2 index: 48 1: (0,1) 2: (1,2)
n: 9 k: 2 index: 49 1: (0,1) 2: (2,2)
n: 9 k: 2 index: 50 1: (1,1) 2: (2,1)
n: 9 k: 2 index: 51 1: (1,1) 2: (0,2)
n: 9 k: 2 index: 52 1: (1,1) 2: (1,2)
n: 9 k: 2 index: 53 1: (1,1) 2: (2,2)
n: 9 k: 2 index: 54 1: (2,1) 2: (0,2)
n: 9 k: 2 index: 55 1: (2,1) 2: (1,2)
n: 9 k: 2 index: 56 1: (2,1) 2: (2,2)
n: 9 k: 2 index: 57 1: (0,2) 2: (1,2)
n: 9 k: 2 index: 58 1: (0,2) 2: (2,2)
n: 9 k: 2 index: 59 1: (1,2) 2: (2,2)
*****
n: 9 k: 3 index: 60 1: (0,0) 2: (1,0) 3: (2,0)
n: 9 k: 3 index: 61 1: (0,0) 2: (1,0) 3: (0,1)
n: 9 k: 3 index: 62 1: (0,0) 2: (1,0) 3: (1,1)
n: 9 k: 3 index: 63 1: (0,0) 2: (1,0) 3: (2,1)
n: 9 k: 3 index: 64 1: (0,0) 2: (1,0) 3: (0,2)
n: 9 k: 3 index: 65 1: (0,0) 2: (1,0) 3: (1,2)
n: 9 k: 3 index: 66 1: (0,0) 2: (1,0) 3: (2,2)
n: 9 k: 3 index: 67 1: (0,0) 2: (2,0) 3: (0,1)
n: 9 k: 3 index: 68 1: (0,0) 2: (2,0) 3: (1,1)
n: 9 k: 3 index: 69 1: (0,0) 2: (2,0) 3: (2,1)
n: 9 k: 3 index: 70 1: (0,0) 2: (2,0) 3: (0,2)

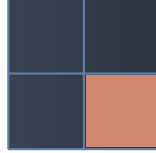
```



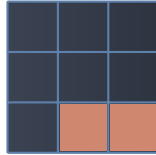
n: 9 k: 3 index: 71 1: (0,0) 2: (2,0) 3: (1,2)  
n: 9 k: 3 index: 72 1: (0,0) 2: (2,0) 3: (2,2)  
n: 9 k: 3 index: 73 1: (0,0) 2: (0,1) 3: (1,1)  
n: 9 k: 3 index: 74 1: (0,0) 2: (0,1) 3: (2,1)  
n: 9 k: 3 index: 75 1: (0,0) 2: (0,1) 3: (0,2)  
n: 9 k: 3 index: 76 1: (0,0) 2: (0,1) 3: (1,2)  
n: 9 k: 3 index: 77 1: (0,0) 2: (0,1) 3: (2,2)  
n: 9 k: 3 index: 78 1: (0,0) 2: (1,1) 3: (2,1)  
n: 9 k: 3 index: 79 1: (0,0) 2: (1,1) 3: (0,2)  
n: 9 k: 3 index: 80 1: (0,0) 2: (1,1) 3: (1,2)  
n: 9 k: 3 index: 81 1: (0,0) 2: (1,1) 3: (2,2)  
n: 9 k: 3 index: 82 1: (0,0) 2: (2,1) 3: (0,2)  
n: 9 k: 3 index: 83 1: (0,0) 2: (2,1) 3: (1,2)  
n: 9 k: 3 index: 84 1: (0,0) 2: (2,1) 3: (2,2)  
n: 9 k: 3 index: 85 1: (0,0) 2: (0,2) 3: (1,2)  
n: 9 k: 3 index: 86 1: (0,0) 2: (0,2) 3: (2,2)  
n: 9 k: 3 index: 87 1: (0,0) 2: (1,2) 3: (2,2)  
n: 9 k: 3 index: 88 1: (1,0) 2: (2,0) 3: (0,1)  
n: 9 k: 3 index: 89 1: (1,0) 2: (2,0) 3: (1,1)  
n: 9 k: 3 index: 90 1: (1,0) 2: (2,0) 3: (2,1)  
n: 9 k: 3 index: 91 1: (1,0) 2: (2,0) 3: (0,2)  
n: 9 k: 3 index: 92 1: (1,0) 2: (2,0) 3: (1,2)  
n: 9 k: 3 index: 93 1: (1,0) 2: (2,0) 3: (2,2)  
n: 9 k: 3 index: 94 1: (1,0) 2: (0,1) 3: (1,1)  
n: 9 k: 3 index: 95 1: (1,0) 2: (0,1) 3: (2,1)  
n: 9 k: 3 index: 96 1: (1,0) 2: (0,1) 3: (0,2)  
n: 9 k: 3 index: 97 1: (1,0) 2: (0,1) 3: (1,2)  
n: 9 k: 3 index: 98 1: (1,0) 2: (0,1) 3: (2,2)  
n: 9 k: 3 index: 99 1: (1,0) 2: (1,1) 3: (2,1)  
n: 9 k: 3 index: 100 1: (1,0) 2: (1,1) 3: (0,2)  
n: 9 k: 3 index: 101 1: (1,0) 2: (1,1) 3: (1,2)  
n: 9 k: 3 index: 102 1: (1,0) 2: (1,1) 3: (2,2)  
n: 9 k: 3 index: 103 1: (1,0) 2: (2,1) 3: (0,2)  
n: 9 k: 3 index: 104 1: (1,0) 2: (2,1) 3: (1,2)  
n: 9 k: 3 index: 105 1: (1,0) 2: (2,1) 3: (2,2)  
n: 9 k: 3 index: 106 1: (1,0) 2: (0,2) 3: (1,2)  
n: 9 k: 3 index: 107 1: (1,0) 2: (0,2) 3: (2,2)  
n: 9 k: 3 index: 108 1: (1,0) 2: (1,2) 3: (2,2)  
n: 9 k: 3 index: 109 1: (2,0) 2: (0,1) 3: (1,1)  
n: 9 k: 3 index: 110 1: (2,0) 2: (0,1) 3: (2,1)  
n: 9 k: 3 index: 111 1: (2,0) 2: (0,1) 3: (0,2)  
n: 9 k: 3 index: 112 1: (2,0) 2: (0,1) 3: (1,2)  
n: 9 k: 3 index: 113 1: (2,0) 2: (0,1) 3: (2,2)  
n: 9 k: 3 index: 114 1: (2,0) 2: (1,1) 3: (2,1)  
n: 9 k: 3 index: 115 1: (2,0) 2: (1,1) 3: (0,2)  
n: 9 k: 3 index: 116 1: (2,0) 2: (1,1) 3: (1,2)  
n: 9 k: 3 index: 117 1: (2,0) 2: (1,1) 3: (2,2)  
n: 9 k: 3 index: 118 1: (2,0) 2: (2,1) 3: (0,2)  
n: 9 k: 3 index: 119 1: (2,0) 2: (2,1) 3: (1,2)  
n: 9 k: 3 index: 120 1: (2,0) 2: (2,1) 3: (2,2)

n: 9 k: 3 index: 121 1: (2,0) 2: (0,2) 3: (1,2)  
n: 9 k: 3 index: 122 1: (2,0) 2: (0,2) 3: (2,2)  
n: 9 k: 3 index: 123 1: (2,0) 2: (1,2) 3: (2,2)  
n: 9 k: 3 index: 124 1: (0,1) 2: (1,1) 3: (2,1)  
n: 9 k: 3 index: 125 1: (0,1) 2: (1,1) 3: (0,2)  
n: 9 k: 3 index: 126 1: (0,1) 2: (1,1) 3: (1,2)  
n: 9 k: 3 index: 127 1: (0,1) 2: (1,1) 3: (2,2)  
n: 9 k: 3 index: 128 1: (0,1) 2: (2,1) 3: (0,2)  
n: 9 k: 3 index: 129 1: (0,1) 2: (2,1) 3: (1,2)  
n: 9 k: 3 index: 130 1: (0,1) 2: (2,1) 3: (2,2)  
n: 9 k: 3 index: 131 1: (0,1) 2: (0,2) 3: (1,2)  
n: 9 k: 3 index: 132 1: (0,1) 2: (0,2) 3: (2,2)  
n: 9 k: 3 index: 133 1: (0,1) 2: (1,2) 3: (2,2)  
n: 9 k: 3 index: 134 1: (1,1) 2: (2,1) 3: (0,2)  
n: 9 k: 3 index: 135 1: (1,1) 2: (2,1) 3: (1,2)  
n: 9 k: 3 index: 136 1: (1,1) 2: (2,1) 3: (2,2)  
n: 9 k: 3 index: 137 1: (1,1) 2: (0,2) 3: (1,2)  
n: 9 k: 3 index: 138 1: (1,1) 2: (0,2) 3: (2,2)  
n: 9 k: 3 index: 139 1: (1,1) 2: (1,2) 3: (2,2)  
n: 9 k: 3 index: 140 1: (2,1) 2: (0,2) 3: (1,2)  
n: 9 k: 3 index: 141 1: (2,1) 2: (0,2) 3: (2,2)  
n: 9 k: 3 index: 142 1: (2,1) 2: (1,2) 3: (2,2)  
n: 9 k: 3 index: 143 1: (0,2) 2: (1,2) 3: (2,2)

és szépen így megy az index a végtelenbe.



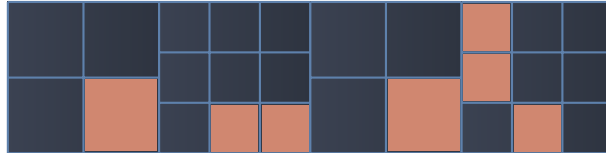
**2.4. Példa.**  $SMNIST(2) = \{(1,0)\}$



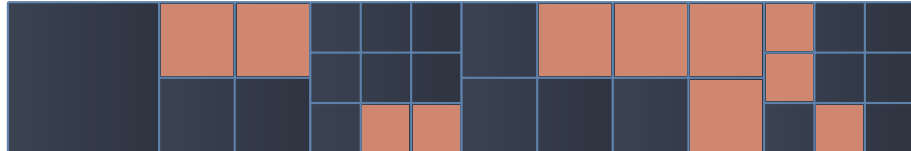
$SMNIST(32) = \{(1,0), (2,0)\}$

Az egerek utálják a macskákat. =

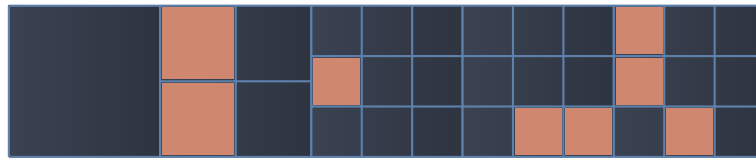
PaRa 2D



4:2:1:1:0:3:2:1:0:2:0:2:1:1:0:3:3:0:2:0:1:1:0

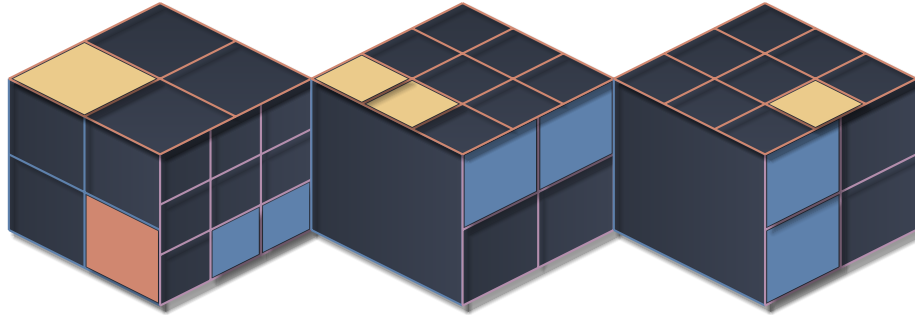


6:1:0:2:2:0:1:1:1:3:2:1:0:2:0:2:1:1:1:2:3:0:1:1:0:1:1:3:3:0:1:  
0:2:1:0



5:1:0:2:2:0:0:0:1:3:1:0:1:3:2:1:0:2:0:3:3:0:2:0:1:1:0

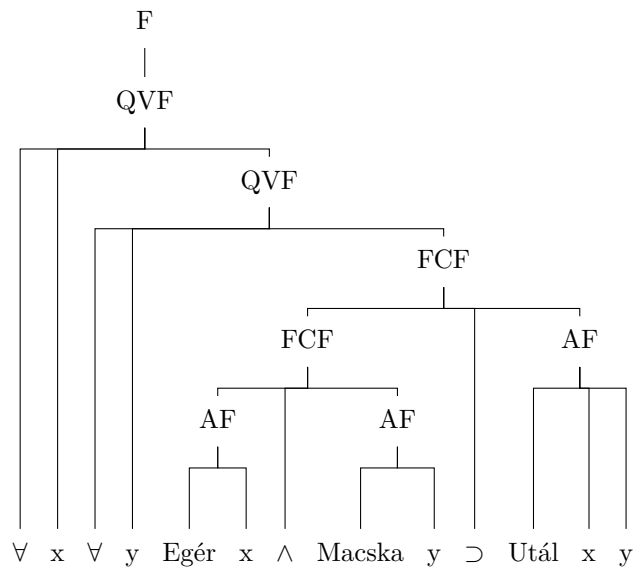
PaRa 3D

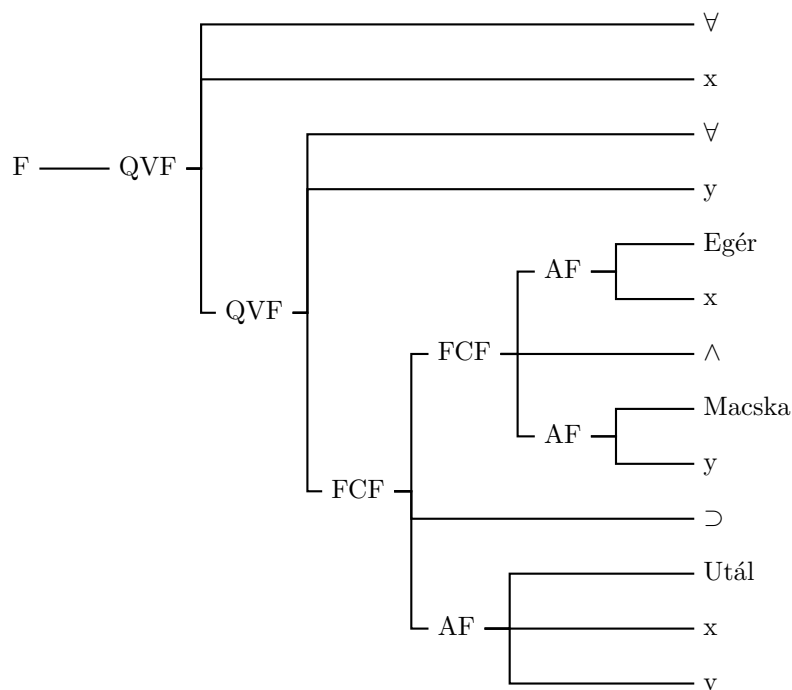


3:2:1:1:0:3:2:1:0:2:0:2:1:1:0:3:3:0:2:0:1:1:0:1:0:1:0:1:0:2:2:  
0:1:1:1:3:2:1:0:2:0:2:1:1:1:2:3:0:1:1:1:1:0:3:3:0:1:0:2:1:0:1:0:2  
:2:0:0:0:1:3:1:0:1:3:2:1:0:2:0:3:3:0:1:0:2:1:0

#### 2.2.5. Inheritance

### 2.3. Second Attempts





## Hivatkozások

- Bátfai, N. (2019a). *Esport culture: a cognitive evolutionary interpretation of artificial intelligence (Esport kultúra: a mesterséges intelligencia kognitív evolúciós értelmezése)*. (Unpublished Manuscript, Original paper in Hungarian [https://gitlab.com/nbatfai/pasigraphy-rhapsody/blob/master/para/docs/para\\_prog\\_guide.pdf](https://gitlab.com/nbatfai/pasigraphy-rhapsody/blob/master/para/docs/para_prog_guide.pdf))
- Bátfai, N. (2019b). *Pasigraphy rhapsody*. <https://gitlab.com/nbatfai/pasigraphy-rhapsody>. GitLab.
- Bátfai, N. (2019c). *Visualization of the language of the esports culture: A preliminary study (Az esport kultúra nyelvének vizualizációja: egy előtanulmány)*. (Bilingual Hungarian and English documentation <https://gitlab.com/nbatfai/pasigraphy-rhapsody/docs>)
- Bátfai, N., Papp, D., Bogacsovics, G., Szabó, M., Simkó, V. S., Bersenszki, M., ... Varga, E. S. (2019). On the notion of number in humans and machines. *CoRR*, *abs/1906.12213*. Retrieved from <http://arxiv.org/abs/1906.12213>
- Dragálin, A., & Buzási, S. (1996). *Bevezetés a matematikai logikába*. Kossuth Egyetemi Kiadó.
- Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (Third ed.). Pearson Education.

## License

% Copyright (C) 2019 Norbert Bátfai

```
% nbatfa@gmail.com, batfai.norbert@inf.unideb.hu
%
% This program is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 3 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with this program. If not, see <https://www.gnu.org/licenses/>.
```