

Automated code generation using

Jazz, the lightweight data processing framework

Open Expo Europe June 2020

Open source software released by BBVA Data & Analytics





- 1. Introduction to Jazz
- 2. The ARC Challenge
- 3. How Big is Big?
- 4. Code Generation in Nature
- 5. Formal Fields
- 6. The Present State of Jazz







Jazz in 2018

- Efficiency is never wrong
- Key-value storage on memory mapped file (LMDB)
- Built-in http client and server (libcurl, μhttpd)
- Highly efficient data structures and containers
- One multithreaded process, the same in every node
- Bebop, a language for the backend (was a project in 2018)





Big Thanks to BBV/\

Releasing Jazz as OSS shows a lot of courage, vision and commitment.

Digital transformation is not just about creating great apps, it is a fundamental change in the relationship with people.





The problem for our solution



Automated code generation







Automated Code Generation

Is NOT: Expressing ideas in a different formal language or expressing them using a diagram and compiling into an executable form.

Introduction to Jazz

IS: Creating a formal (unambiguous, executable) piece of code to solve a problem. The problem can be expressed in human language (ambiguous, inconsistent, assuming prior knowledge, etc.)





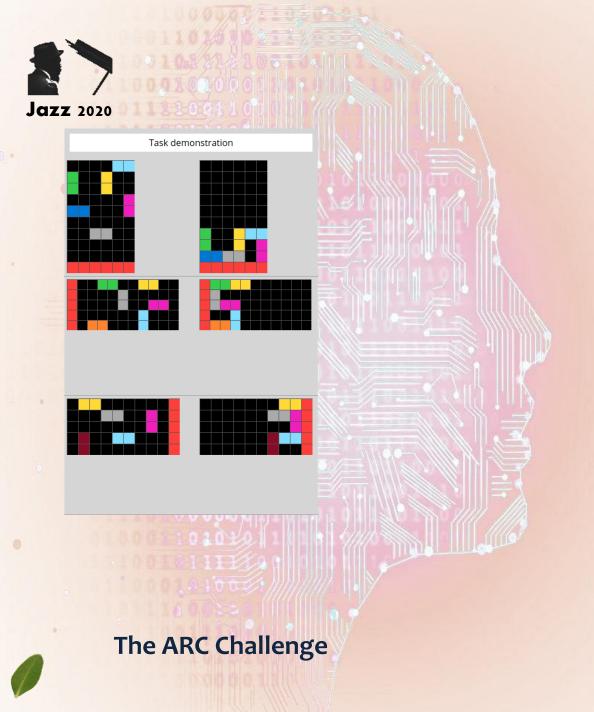
Some recent results

Title	Author, Year	Affiliation
Neural Turing Machines, (+ more)	Graves++, 2014-2018	Google DeepMind, London, UK
Reinforcement Learning Neural Turing Machines	Zaremba & Sutskever, 2015-2019	Facebook AI Research & Google Brain
Stoke (5 papers)	Schkufza++, 2013-2016	Stanford
Prose Team (2 papers)	Gulwani++, 2011-2018	Microsoft Research
Sketch (3 papers)	Solar-Lezama, 2008-2013	MIT
many more	many 2011-2020	Dawn Song (Berkeley), Quoc Le (Google Brain), Reed & de Freitas (Google DeepMind), Joulin (Facebook AI Research), Yale- NUS (Yale), UTOPIA Research Group (University of Texas),



- 1. Introduction to Jazz
- 2. The ARC Challenge
- 3. How Big is Big?
- 4. Code Generation in Nature
- 5. Formal Fields
- 6. The Present State of Jazz

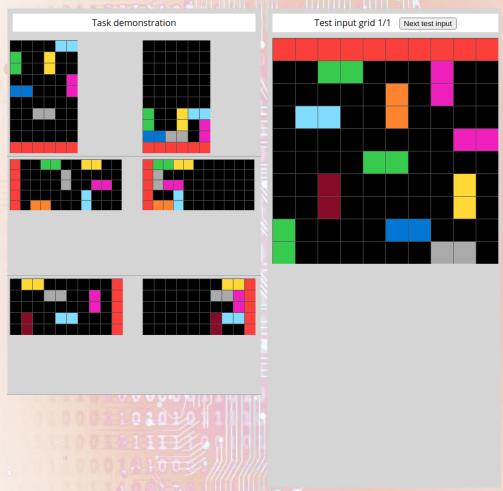






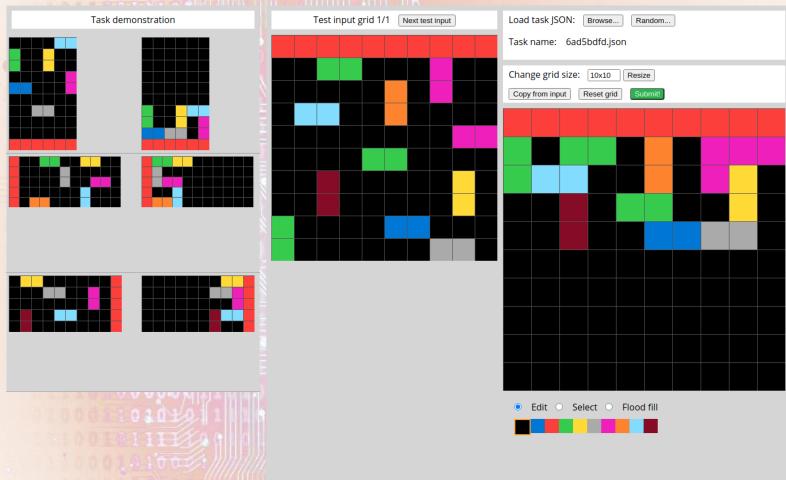












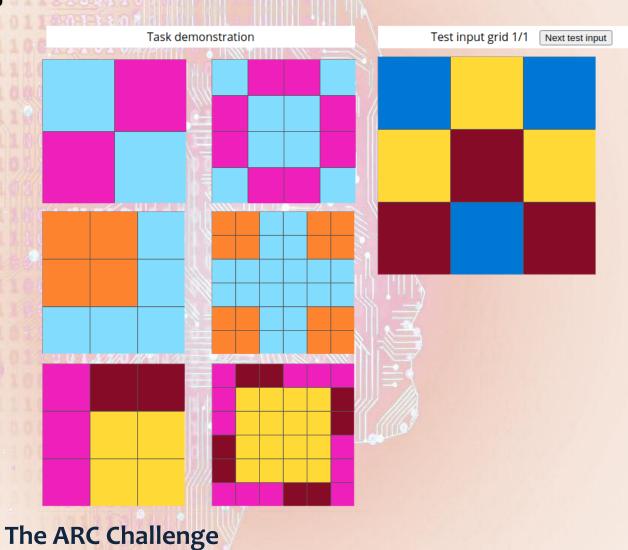






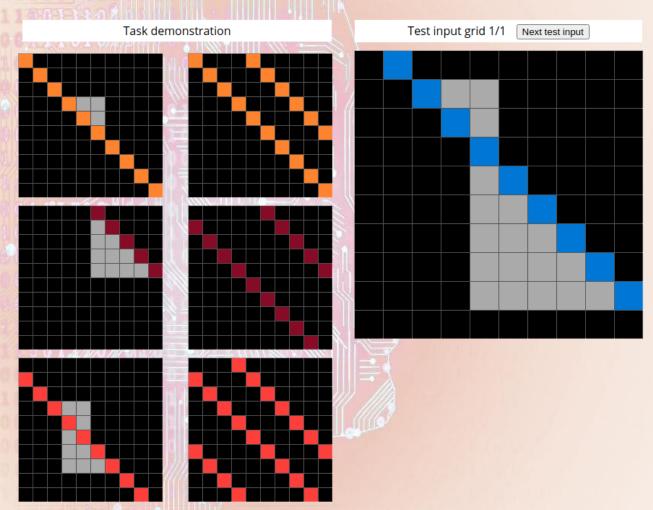






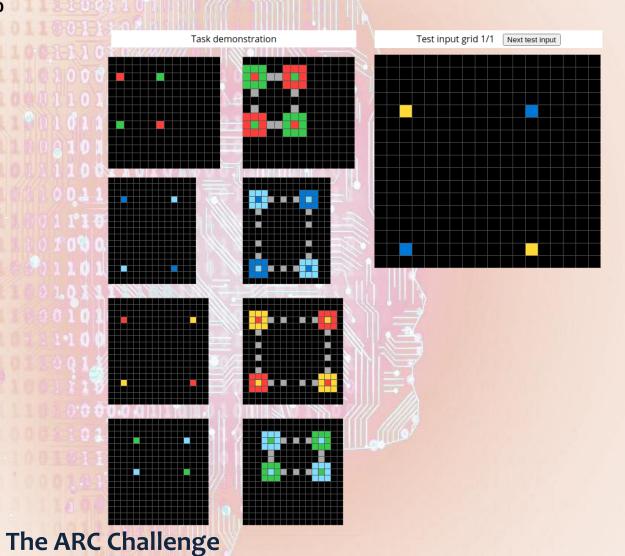






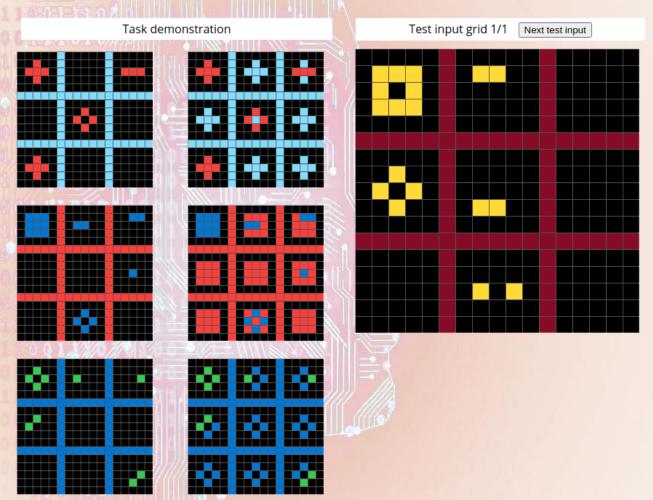






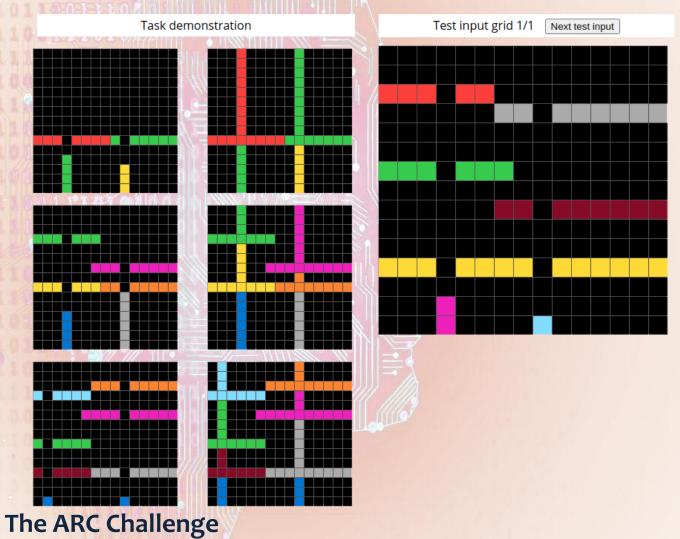






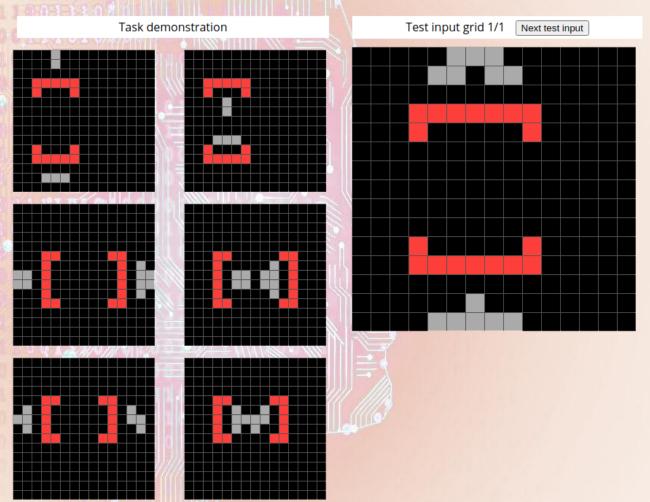






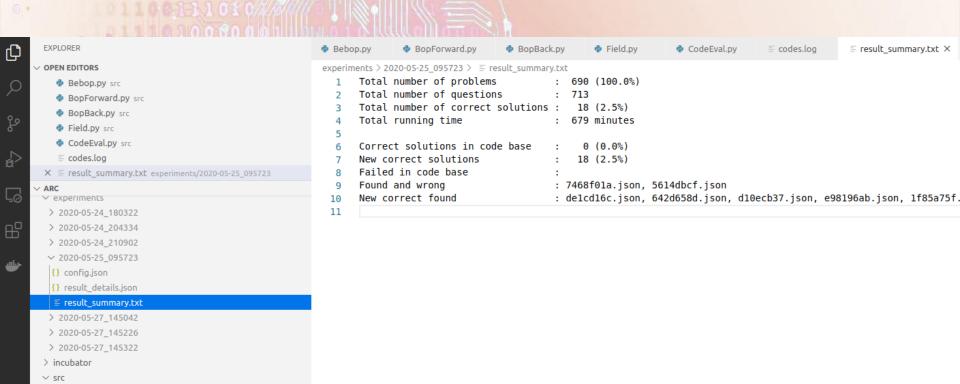














Jazz 2020



```
EXPLORER
                                                                                      Bebop.py
                                                                                                              BopForward.py
                                                                                                                                              BopBack.py
                                                                                                                                                                        Field.py
                                                                                                                                                                                                 CodeEval.py
                                                                                                                                                                                                                             = result_summary.txt
 OPEN EDITORS

≡ codes.log

                                                                                                 ./experiments/2020-05-25 095723
     Bebop.py src
                                                                                          2
     BopForward.py src
                                                                                          3
                                                                                                     Found and wrong
     BopBack.py src
                                                                                          4
     Field.py src
                                                                                                        7468f01a.json
                                                                                          5
     CodeEval.pv src
                                                                                          6
  X ≡ codes.loa
                                                                                                           10.00 ['get question', 'pic fork by color as pics', 'pics as 2pic', 'pic all as pat', 'pat flip left right',
                                                                                          7
                                                                                          8
                                                                                                            0.87 ['get_question', 'pic_fork_by_color_as_pics', 'pics_as_2pic', 'pic_all_as_pat', 'sto_a', 'pat_flip_lef
      Fresult summary.txt experiments/2020-05-25 095723
                                                                                          9
                                                                                                            0.78 ['get question', 'pic fork by color as pics', 'pics as 2pic', 'pic all as pat', 'sto a', 'pat flip up
 ARC
                                                                                        10
                                                                                        11
                                                                                                        5614dbcf.json
 > diagrams
                                                                                        12

∨ experiments

                                                                                                           10.00 ['(3, 3)', 'get question', 'pic intp zoom fit', '(5, 0)', 'swap top2', 'pic intp recolor']
                                                                                        13
  > 2020-05-24 180322
                                                                                                            0.67 ['(3, 3)', 'get question', 'pic intp zoom fit']
                                                                                        14
                                                                                        15
                                                                                                             0.67 ['get question', 'pic fork on v axis as pics', 'pics as 2pic', '(3, 3)', 'get question', 'pic intp zoo
  > 2020-05-24_204334
                                                                                        16
  > 2020-05-24_210902
                                                                                        17
                                                                                                     New correct found
  2020-05-25_095723
                                                                                        18
                                                                                        19
                                                                                                        de1cd16c.json
    {} result details.json
                                                                                        20
    2.50 ['[[1, 2], [2, 1]]', '(6,)', 'int black box as pic', 'swap top2', '(2,)', 'swap top2', 'pic int zoom (
                                                                                        21
  > 2020-05-27_145042
                                                                                                             1.50 ['get question', 'pic fork on auto grid as pics', 'pics main color as vec', 'vec row as pic']
                                                                                        22
                                                                                                             1.50 ['get question', 'pic fork on v axis as pics', 'pics as 2pic', 'get question', 'pic fork on auto grid
                                                                                        23
  > 2020-05-27 145226
                                                                                        24
  > 2020-05-27 145322
                                                                                        25
                                                                                                        642d658d.json
                                                                                        26
                                                                                        27
                                                                                                             6.00 ['get question', 'pic fork on auto grid as pics', 'pics main color as vec', 'vec row as pic', '(2, 3)'
                                                                                                             6.00 ['get question', 'pic fork on auto grid as pics', 'pics main color as vec', 'vec row as pic', '(2, 3)'
                                                                                        28
                                                                                                             3.33 ['[[1, 2], [2, 1]]', '(6,)', 'int black box as pic', 'swap top2', '(2,)', 'swap top2', 'pic int zoom c
                                                                                        29
                                                                                        30
                                                                                                        d10ecb37.json
                                                                                        31
                                                                                        32

✓ TIMELINE codes.log

                                                                                                           10.00 ['get question', 'pic outline 4n', '(2, 2)', '(1, 1)', 'get question', 'pic 2intp crop', '2pic tile al
                                                                                        33
 Adds log from experiments kaalam
                                                                                                            0.42 ['(2, 2)', '(1, 1)', 'get question', 'pic 2intp crop', 'pic all as pat', 'pat flip up down', 'pat as p
                                                                        2 wks
                                                                                        34
                                                                                                             0.42 ['(2, 2)', '(1, 1)', 'get question', 'pic 2intp crop', 'pic all as pat', 'pat flip up down', 'pat as pat', 'pat', 'pat',
                                                                                        35
                                                                                        36
                                                                                                        e98196ab.json
                                                                                        37
                                                                                        38
                                                                                        39
                                                                                                           10.00 ['get question', 'pic fork on h axis as pics', 'pics as 2pic', '2pic maximum']
                                                                                                           10.00 ['(1, 6)', 'get question', 'pic fork on h axis as pics', 'pics as 2pic', '2pic maximum']
                                                                                        40
                                                                                        41
                                                                                                           10.00 ['get question', 'pic fork on h axis as pics', 'pics as 2pic', '2pic maximum', '(6, 0)', 'swap top2',
                                                                                        42
                                                                                        43
                                                                                                        1f85a75f.ison
                                                                                        44
                                                                                                           10.00 ['get question', 'pic fork by color as pics', 'pics filter single color', 'pics as pic']
            The ARC Challenge
```



- 1. Introduction to Jazz
- 2. The ARC Challenge
- 3. How Big is Big?
- 4. Code Generation in Nature
- 5. Formal Fields
- 6. The Present State of Jazz





Class 1 (of 7): Macroscopic



	Human population			
	Human population		7.8 x 10 ⁹	
	Doing something 1000 times (taking pictures, owning money, eating,)		7.8 x 10 ¹²	
	Doing something a million times		7.8×10^{15}	
GDP of the whole planet		8.6 x 10 ¹³ (USD)		
Age of the universe		1.4×10^{10} (years) 4.3×10^{17} (sec)		
Weight of all biomass of the planet		5.6 x 10 ¹¹ (ton) 5.6 x 10 ¹⁴ (Kg)		

Computer resources by the €	Clock Cycles	Evaluations @(1M/sec)
1 euro	4 x 10 ¹³	4 x 10 ¹⁰
1 million euros	4 x 10 ¹⁹	4 x 10 ¹⁶
1 billion euros	4 x 10 ²²	4 x 10 ¹⁹



Class 2 (of 7): The Universe openexpo





Number of particles in the universe: 1080







Class 3 (of 7): Combinatorial

- Do a sequence of decisions taken from a limited set: (choose words, play games, type text, do complex manipulation, ...)
- Fit a large model
- Generate images
- Code

•





Class 4 (of 7): Formal



```
1 * nest <- function(func, arg, times) {</pre>
3
      for (i in 1:times) ret <- func(ret)
5
      ret
   A <- function(n) nest(factorial, n, n)
10
   A(1) = 1! = 1
11
   A(2) = 2!! = 2
   14
              6415651620520158737919845877408325291052446903888118841237643411919510455053466586162432719401971139098455367272785370993456\\
15
              2985558671936977407000370043078375899742067678401696720784628062922903210716166986726054898844551425719398549944893959449606
16
                                                                                                          4628
                   A(4) = 4!!!! cannot be computed on Earth
              824234
                                                                                                          3601
17
18
              294475
                                                                                                          9318
                    B(9) = A(A(A(A(A(A(A(A(A(9))))))))) cannot be computed using the whole
19
                                                                                                          3803
20
              056825
                                                                                                          3757
21
              247722
                          universe as a computer
                                                                                                          2335
22
              630992
                                                                                                          7756
23
                    C(9) = B(B(B(B(B(B(B(B(9))))))))) is still a finite number
                                                                                                          5666
24
                                                                                                          5168
25
26
              27
              00000000000
28
29
   A(4) = 4!!!! cannot be computed on Earth
30
31
   And you can continue ...
32
33
   B <- function(n) nest(A, n, n)
34
35
   B(9) = A(A(A(A(A(A(A(A(A(9)))))))))) cannot be computed using the whole universe as a computer
36
   C <- function(n) nest(B, n, n)
37
38
   C(9) = B(B(B(B(B(B(B(B(9)))))))) is still a finite number
```



Classes (5, 6, 7):



Countable, Continuum and the Power Set of R

- No
- Countable:

- The set of all integers
- The set of rational numbers
- The set of tuples of integers

N

Continuum:

- The set of all real numbers
- The set of tuples of real numbers

N

Power Set of R:

- The set of all sets of real numbers
- The set of functions from R → R



- 1. Introduction to Jazz
- 2. The ARC Challenge
- 3. How Big is Big?
- 4. Code Generation in Nature
- 5. Formal Fields
- 6. The Present State of Jazz







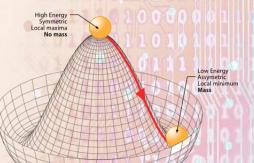
Combinatorial and Formal

- Both classes are bigger than anything material, including the universe. Only trivial combinatorial problems are small enough to be "brute forced" (e.g., tic-tac-toe).
- We know how to search problems in the class of combinatorial problems, given there is some structure and we are not searching for a "needle in a haystack".
- We don't know how to search the class of formal problems. Without some limitation, it is "just too big".



The origin of everything (before code)

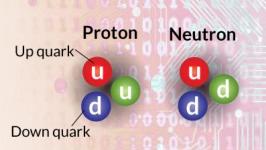


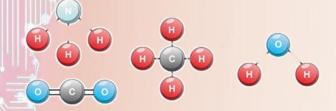


Energy - Control of the control of t

3. Atoms: Nuclear Physics, 1906-1950s

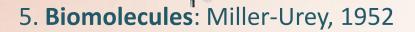
1. Matter: Higgs mechanism, 1960-2012





4. Molecules: Chemistry XIX (1869, Mendeleev)

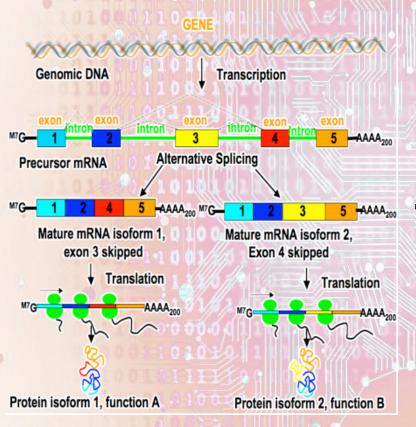
2. Protons, e: Particle Physics, 1950-1970s



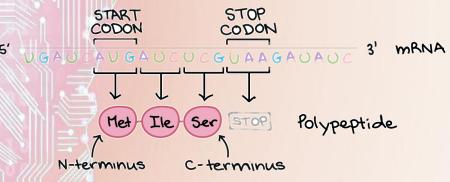


The origin of everything (code)





We understand how code works in nature, but ...



... we don't know how, when or even where it originated.



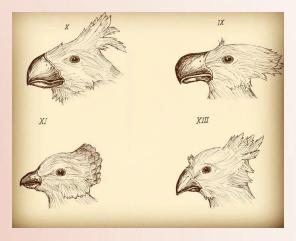
The origin of everything





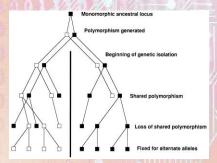
(after code)

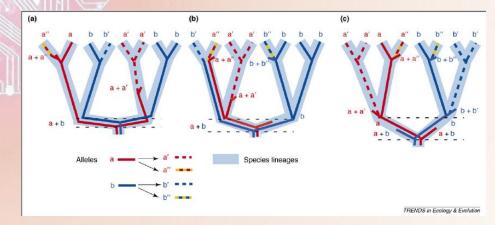




Lamarck & Darwin XIX: "Theory of Evolution",

Watson & Crick, 1953 + others: Science of Evolution









Important Ideas

- 1. If automatic code creation was impossible, we would not exist.
- 2. It is not as hard as solving **abiogenesis**. We don't create code out of nothing, we:
 - Copy existing short code items
 - Mutate code items
 - Fit arguments to existing code items
 - Recombine items to form new snippets



Forms of code (1 of 2)



```
#include <opencv2/opencv.hpp>
#include <iostream>
using namespace std;
using namespace cv;
int main(int, char** argv)
   // Load the image
    Mat src = imread(argv[1]);
   // Check if everything was fine
    if (!src.data)
         return -1;
   // Show source image
    imshow("Source Image", src);
   // Change the background from white to black, since that will help later to
     extract
   // better results during the use of Distance Transform
    for( int x = 0; x < src.rows; x++ ) {
       for( int y = 0; y < src.cols; y++ ) {
            if ( src.at<Vec3b>(x, y) == Vec3b(255,255,255) ) {
              src.at<Vec3b>(x, y)[0] = 0;
              src.at<Vec3b>(x, y)[1] = 0;
              src.at < Vec3b > (x, y)[2] = 0;
   // Show output image
    imshow("Black Background Image", src);
 Figure 13.
at - organ No. 1 of Figure 10:
                                                                (a) Nearest ( ) and next nearest ( )
                                                                      neighbors of X
     - organ No. 2 of Figure 10:
       - organ No. 3 of Figure 10:
                                                                            (c) Nearest ( 🔾 ) and next near-
                                                                              est ( ) neighbors of X
                                                                    Flg. 4. The quadratic lattice
```

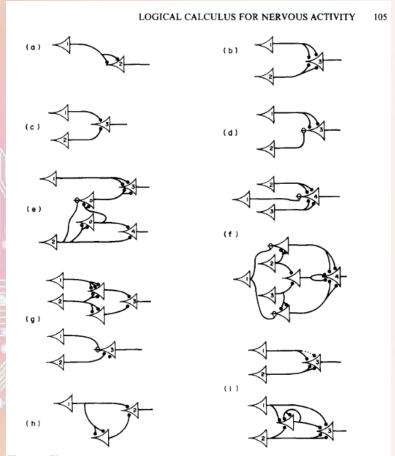
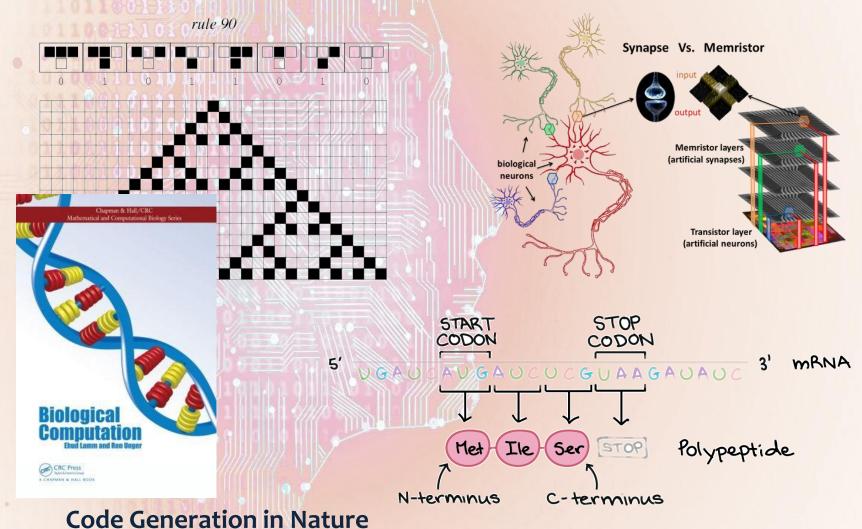


Figure 1. The neuron c_i is always marked with the numeral i upon the body of the cell, and the corresponding action is denoted by "N" with i's subscript, as in the text:



Forms of code (2 of 2)







Takeaways from code in Nature



- 1. Code is a sequence run once
 - It can stop anytime (error is a result)
 - Conditionals by "inhibition" (arguments, not jumps)
- 2. Code has structure
 - Primary structure (opcodes & types)
 - Secondary (items are evaluated)
 - Tertiary (snippets have a goal)



- 1. Introduction to Jazz
- 2. The ARC Challenge
- 3. How Big is Big?
- 4. Code Generation in Nature
- 5. Formal Fields
- 6. The Present State of Jazz





Putting it all Together



- In general, code generation belongs to the class of formal problems. Even if some problems in that class can be written simply (low Kolmogorov complexity), anything in that class is "just too big" and non computable.
- Nature solves this problem by creating code that executes in sequence just once and does conditionals through inhibition or in translation.
- Accepting some (apparent) limitations we can still have Turing-complete code snippets and search them in the class of not too big combinatorial problems with feedback and structure.

Formal Fields



Formal Fields



Is a framework to automate code generation across domains using the same algorithms and language grammar.

Kind: A set of types used in source and destination

Formal field: A source and destination + a domain language

Relation: A field + a code base

Prior: Value of code item from previous experience

Evaluation: A vector with intermediate goals for an item

Reward: A value based on evaluation used in search

Formal fields are intended to enable multi-domain lifelong learning systems.

Formal Fields



Bebop



Level	term	description
Primary structure	opcodes	built-in functions
Secondary struct.	items	shortest seq. that can be evaluated
u	alleles	items with same code and different args.
u	isomorphisms	items with same type seq. (form)
Tertiary struct.	snippets	complete programs source → dest.
и	code base	collection of working snippets

- One time sequence of opcodes
- Strictly typed
- Express complex domains and functions: speech, video, ...
- Break as fast as possible (HCF is a result)
- Do conditionals via arguments
- Do loops by rewarding repetition

HALT AND CATCH FIRE

HCF is a core opcode.

Formal Fields



The Formal Fields Paper



JazzARC

Proof of Concept Code Generation using Formal Fields applied to the ARC Challenge

build passing python 3.7 License GPL Codecov 100% Maintained? no Ask me anything







Formal Fields paper and Open Expo slides

We will update this project with links to:

- · Formal Fields Paper
- Open Expo 2020 presentation call
- Open Expo 2020 presentation (slides)
- Stav tuned!

License

Note that even if Jazz has an Apache license, this is GPLed. The way to go is Jazz, this is research-only material. Of course, the content of the data folder belongs to @fchollet, is under Apache License 2.0 and can be found here.

This is a PoC of the Jazz platform in Python to research automated code generation. The classes with a Thelonious icon are Jazz classes with the same name (simplified for the PoC)



Will be linked in the github repository: https://github.com/kaalam/JazzARC





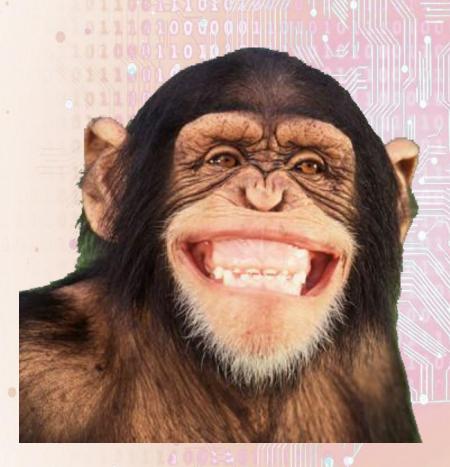
- 1. Introduction to Jazz
- 2. The ARC Challenge
- 3. How Big is Big?
- 4. Code Generation in Nature
- 5. Formal Fields
- 6. The Present State of Jazz





We are thrilled !!





- 1. We found the problem for our solution
- 2. We experience intelligence emerging from a machine
- 3. We have many years ahead tackling exciting problems while building Open Source Software

The Present State of Jazz



Jazz is more important than ever !!!





1. Going from proof of concept to a production level, most efficient possible, scalable process.

2. Building code, knowledge and a team to last many years.

The Present State of Jazz





Now, we are on our own, with our strengths ...

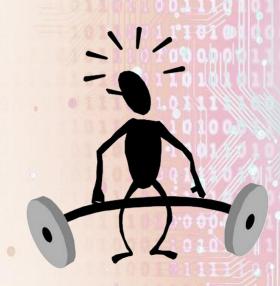


- We don't just shoot at the moon, we shoot at a moon in a distant galaxy.
- We have decades of successful experience, in AI to understand what works and why.
- We deliver.
- We stand on the shoulders of giants. OSS
- •We are used to wearing the running shoes.
- We enjoy doing it.





... and weaknesses



- We still have to finish the MVP.
 - We have to create a community.
- We need success stories.
- Short version: We need help.





And remember, the product is the second most important thing in any Open Source Software Project





The most important thing in any Open Source Software Project is, of course ...





The community







We need volunteers!





Thankyou!

ARC challenge: https://github.com/kaalam/JazzARC

Development: https://github.com/kaalam/Jazz

Programming doc: https://kaalam.github.io/develop

Jazz reference: https://kaalam.github.io/jazz_reference

kaalam.ai

@kaalam_ai

