Synthesis of Recursive Functions from Sequences of Natural Numbers[1]

Thibault Gauthier

September 8, 2021

[1] Supported by the Czech Science Foundation project 20-06390Y

Problem

Given a "finite" sequence of natural numbers,

$$A000217: \quad 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120$$

generate a "small" program that match the sequence?

$$f(x) = \frac{x \times (x+1)}{2}$$

Motivations

1) Conjecturing:

$$\sum_{x=0}^{n} x = \frac{x \times (x+1)}{2} \quad ?$$

2) A shorter explanation generalizes better.

3) A shorter explanation gives some understanding.

$$\sum_{x=0}^{n} x \text{ is not prime for } x > 2$$

Language: understandable, efficient, generalizes

one variable: x

constants: 0, 1, 2

functions: $+, -, \times, /, mod, \sqrt{\ }, power$

conditional statements:
- $cond(a, b, c) =$ if $a = 0$ then b else c
- $loop(f, a, b) = f^a\ (b)$
- $halt(f, a) =$ minimum i such that $f^i(a) = 0$

A issue with linear synthesis?

Same sub-expression repeatedly synthesized.

$$f(x) \times (x \times (x + 1)) \times g(x)$$

$$(x \times (x + 1)) + h(x)$$

Factorized bottom-up synthesis

target T: $[0,\ 1,\ 3,\ 6,\ 10,\ 15, \ldots]$

size 1: $x,\ 0,\ 1,\ 2$

size 2: $\sqrt{x},\ \sqrt{0},\ \sqrt{1},\ \sqrt{2}$

size 3: $x + x,\ x + 1,\ 2 \times x,\ \ldots$

size 4: $loop(\lambda x.\ x,\ 1,\ 2), \sqrt{x + x}, \ldots$

solution: program f such as $[f(0),\ f(1), \ldots, f(15)] = T$

random fixed width $w \in \{4, 8, 16, 32\}$ for each search:
select w programs at each size.

Selecting programs for a target

target: $[0,\ 1,\ 3,\ 6,\ 10,\ 15, \ldots]$

sub-program: $x + 1 \equiv [1,\ 2,\ 3,\ 4,\ 5,\ 6 \ldots]$

$[1,\ 2,\ 3,\ 4,\ 5,\ 6 \ldots]$ useful for $[0,\ 1,\ 3,\ 6,\ 10,\ 15, \ldots]$?

Euclidean distance is not good

because $[2, 14, 3, \ldots]$ is useful for $[2^2,\ 2^{14},\ 2^3, \ldots]$.

Train a classifier from solutions

Let P be a minimal solution and P_{pos} a subprogram of P.

positive example: $([P], [P_{pos}])$

Let P_{neg} be a generated program with the same size as P_{pos} which is not a subprogram of P.

negative example: $([P], [P_{neg}])$

OEIS sequences and restrictions

1) At least 16 elements.

2) First 16 elements between 0 and $2^{63} - 1$.

3) First 8 elements different form every other OEIS sequence.

About 350 000 sequences become about 200 000 targets.

Reinforcement learning

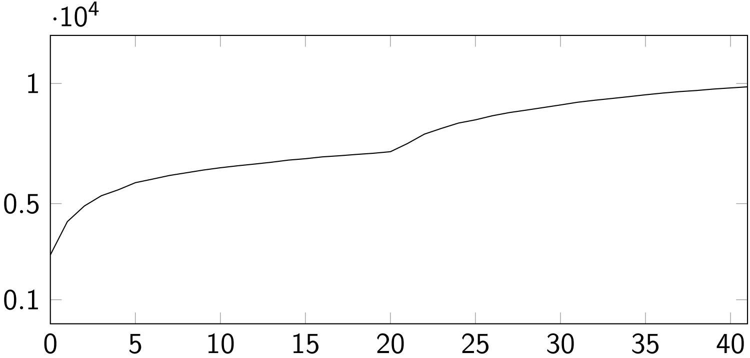

Figure: Number y of training problems solved after generation x

Solutions

Picked at random: A079273
Octo numbers

$$1, 10, 29, 58, 97, 146, 205, 274, 353, 442, \ldots$$

$$(x \times x) + (1 + (x + x))^2$$

Smallest with a nested loop: A125833
Numbers whose base 5 representation is 333.......3

$$0, 3, 18, 93, 468, 2343, 11718, 58593, 292968, \ldots$$

Def: $f(x) = 1 + (x + x), \quad g(x) = x + f^2(x)$

$$g^x(0)$$

Solutions

Largest: A273848
Number of active (ON,black) cells at stage $2^n - 1$ of the two-dimensional cellular automaton defined by "Rule 969", based on the 5-celled von Neumann neighborhood.

$$1, 4, 45, 225, 961, 3969, 16129, 65025, 261121, \ldots$$

Def: $f(x) = x + (1 + x)$

$$(\text{if } x/2 = 0 \text{ then } 2^x \text{ else if } \sqrt{x+1}/2 = 0 \text{ then } 3^x \text{ else } f^x(1))$$

$$\times$$

$$(\text{if } x/2 = 0 \text{ then } 2^x \text{ else if } \sqrt{x+1}/2 = 0 \text{ then } f(x) \text{ else } f^x(1))$$

Conclusion

Factorized bottom-up program synthesis

Semantic quotient + semantic filtering (semantic = sequences)

Interesting results (requires more learning)

Future work

More extensive experiments:
- mix syntactic and semantic features
- large integers, real numbers, multiple variables, lists
- **backward reasoning, recursion**, techniques from ILP

Apply synthesis to theorem proving:
- term synthesis, tactic synthesis, cut introduction.

Look for applications beyond mathematics.