FlexCoder: Practical program synthesis with flexible input lengths and expressive lambda functions

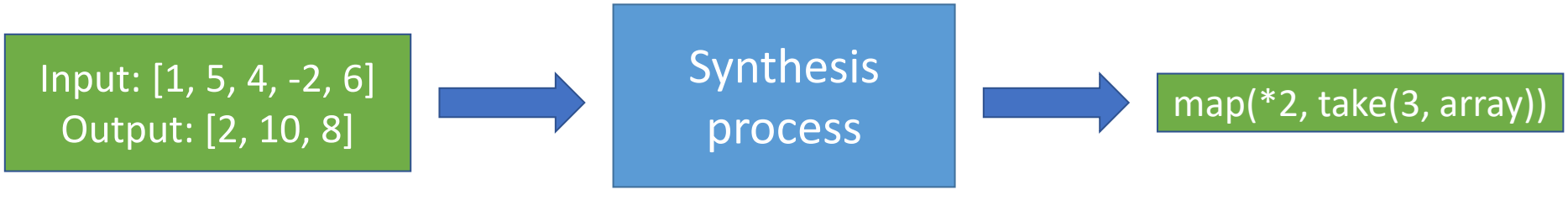
Bálint Mucsányi Bálint Gyarmathy Ádám Tibor Czapp
Dávid Szilágyi Balázs Pintér

ELTE IK

February 5, 2021

Introduction

Synthesis process



Function composition

Introduction

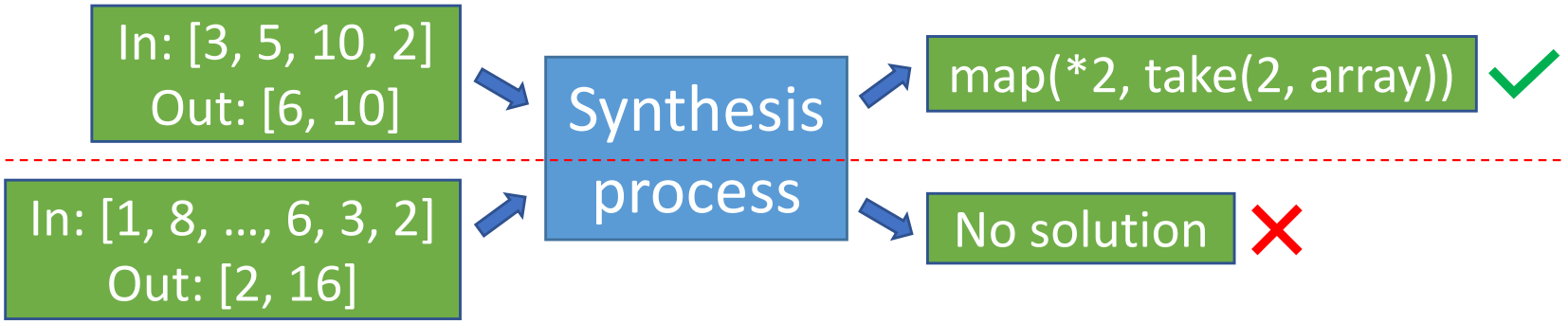
- Task
  - Synthesizing programs based on input-output pairs
- Essential parts:
  - Grammar and data generation
    - Deep learning algorithm $\Rightarrow$ data-hungry approach
  - Neural network
  - Search algorithm

Components, Limitations

How do the components work together?

Main limitations of state-of-the-art systems:

- static or upper-bounded input vector sizes



- agglutination of grammar tokens
  - map($*2$, list)
- limited parameter ranges of lambda operators
- limited integer ranges of inputs

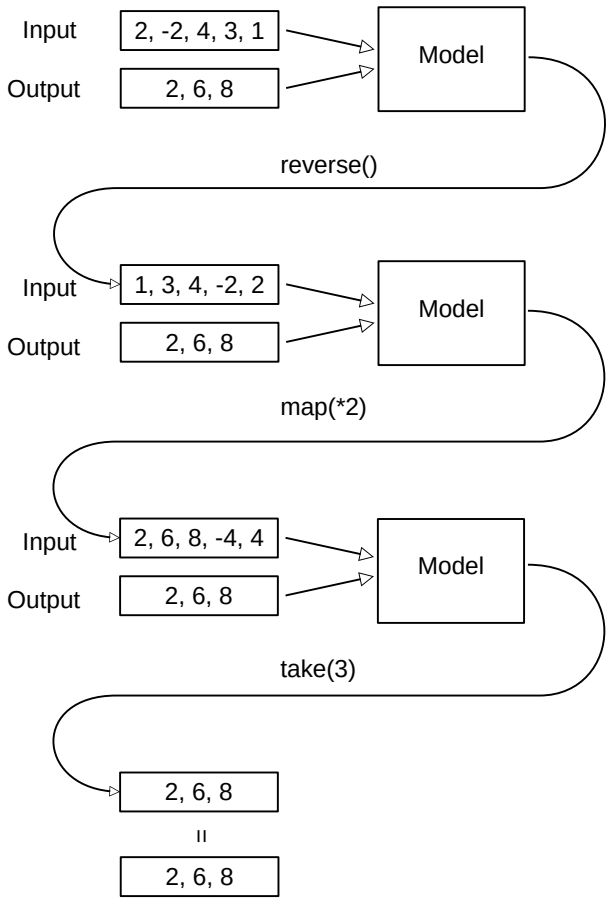
Grammar

A simplified version of the grammar

$$
\begin{aligned}
S &\to ARRAY\_FUNCTION \mid NUMERIC\_FUNCTION \\
ARRAY\_FUNCTION &\to take(NUM,\ ARRAY) \mid drop(NUM,\ ARRAY) \\
ARRAY\_FUNCTION &\to map(LAMBDA\_FUNCTION,\ ARRAY) \\
ARRAY\_FUNCTION &\to filter(LAMBDA\_FUNCTION,\ ARRAY) \mid \dots \\
NUMERIC\_FUNCTION &\to max(ARRAY) \mid min(ARRAY) \\
NUMERIC\_FUNCTION &\to sum(ARRAY) \mid count(ARRAY) \mid \dots \\
NUM &\to -8 \mid \dots \mid 8 \\
LAMBDA\_FUNCTION &\to OPERATOR\ NUM \\
OPERATOR &\to * \mid / \mid + \mid \% \mid > \mid == \mid \dots \\
ARRAY &\to list \mid ARRAY\_FUNCTION
\end{aligned}
$$

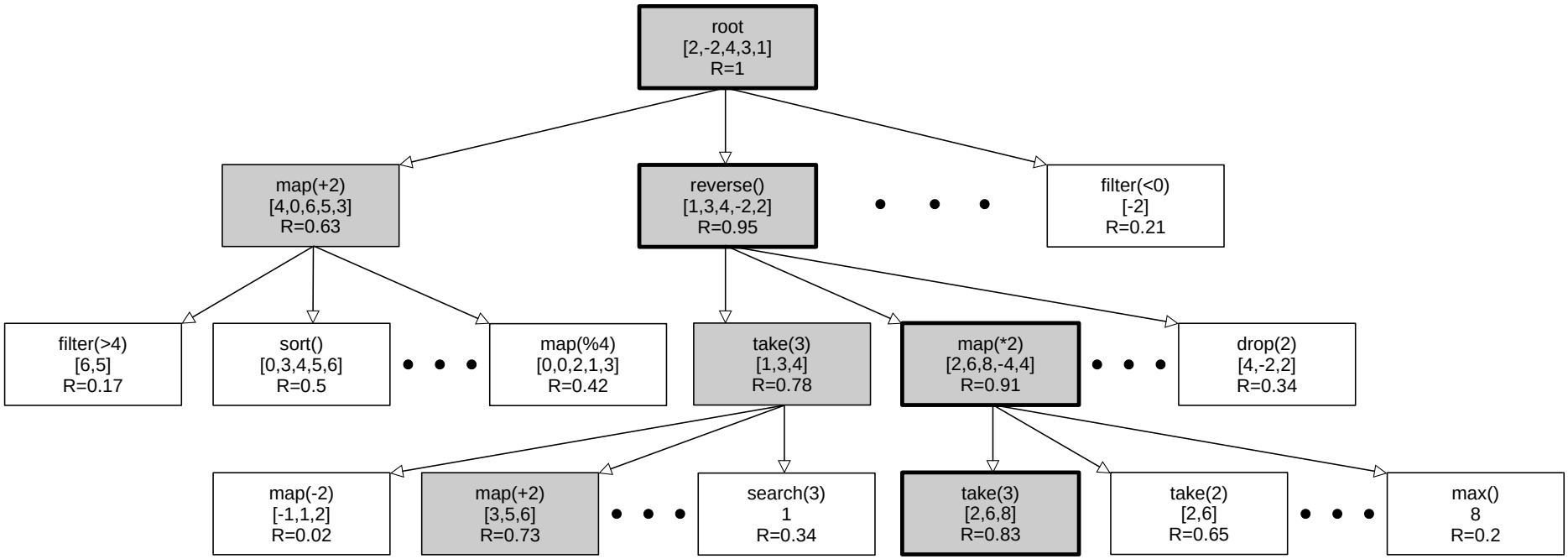
$$
\begin{aligned}
S &\to NUMERIC\_FUNCTION \to max(ARRAY) \to max(ARRAY\_FUNCTION) \\
&\to max(map(LAMBDA\_FUNCTION, ARRAY)) \to max(map(*2, list))
\end{aligned}
$$

Synthesizing Process



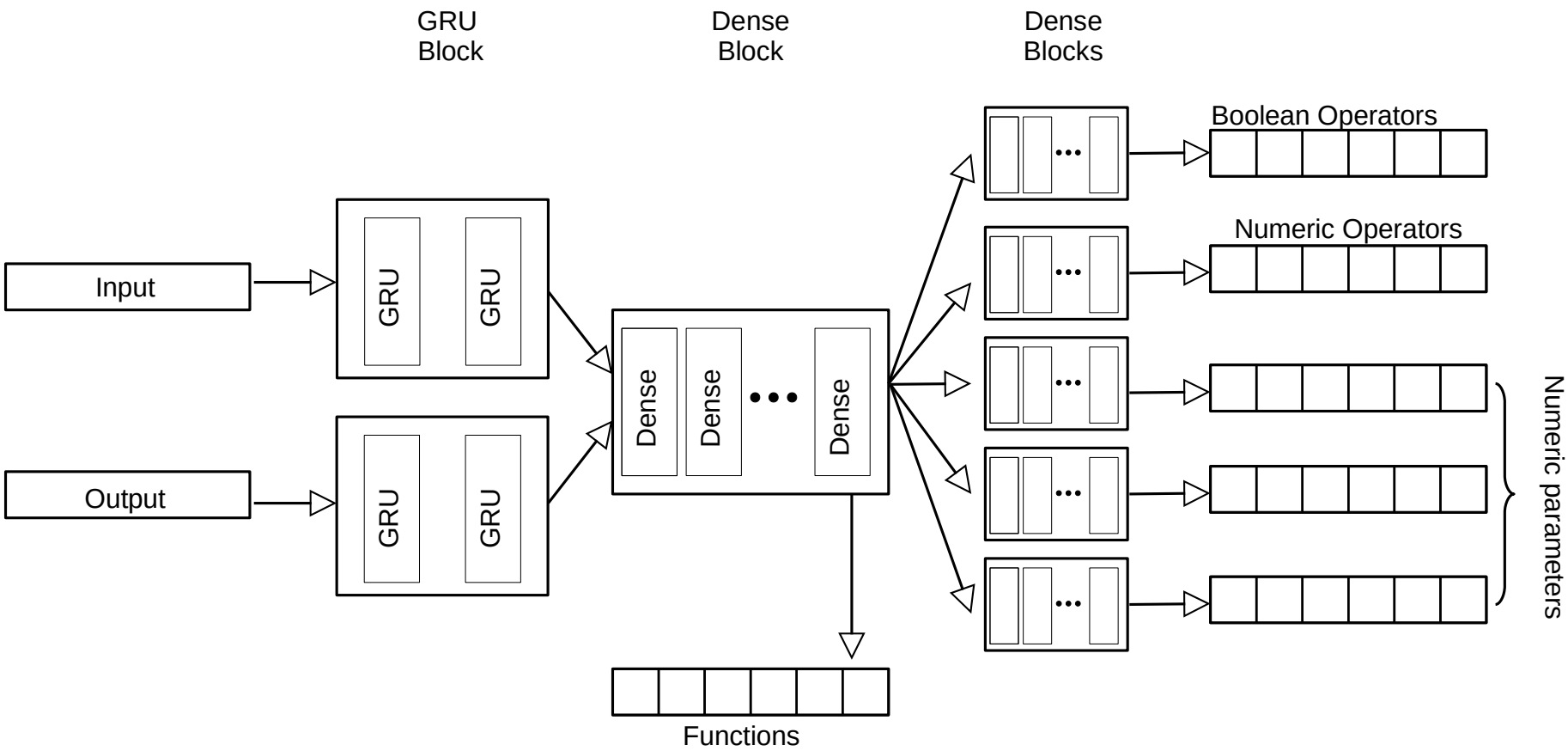
- The input(s) and corresponding output(s) are given
- The composition is built one function at a time
- The neural network guides the search as a heuristic.
- take(3,map(*2,reverse(inp)))

Search

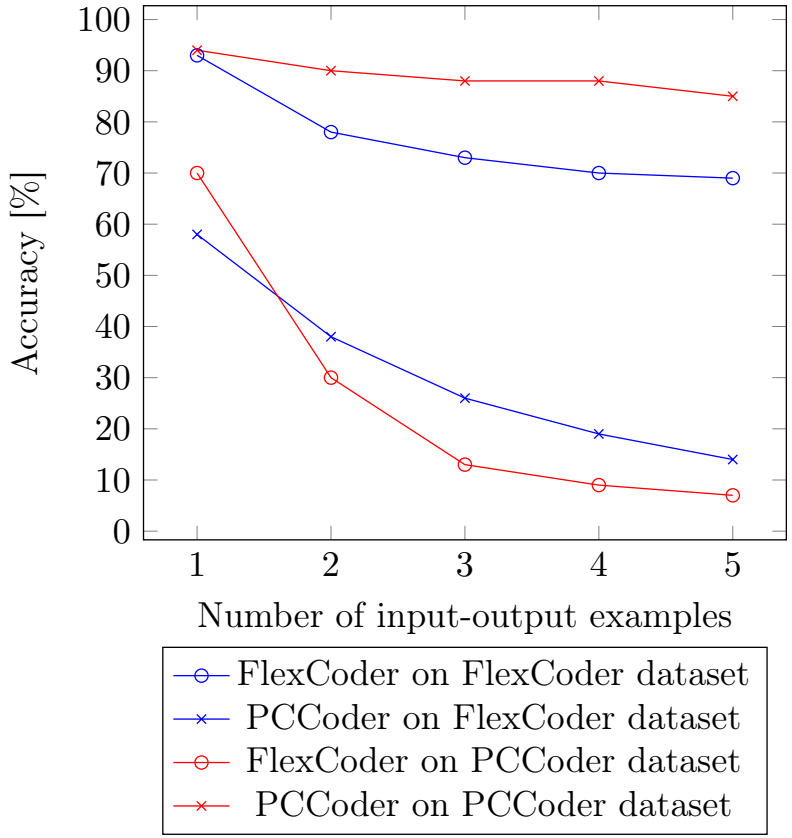


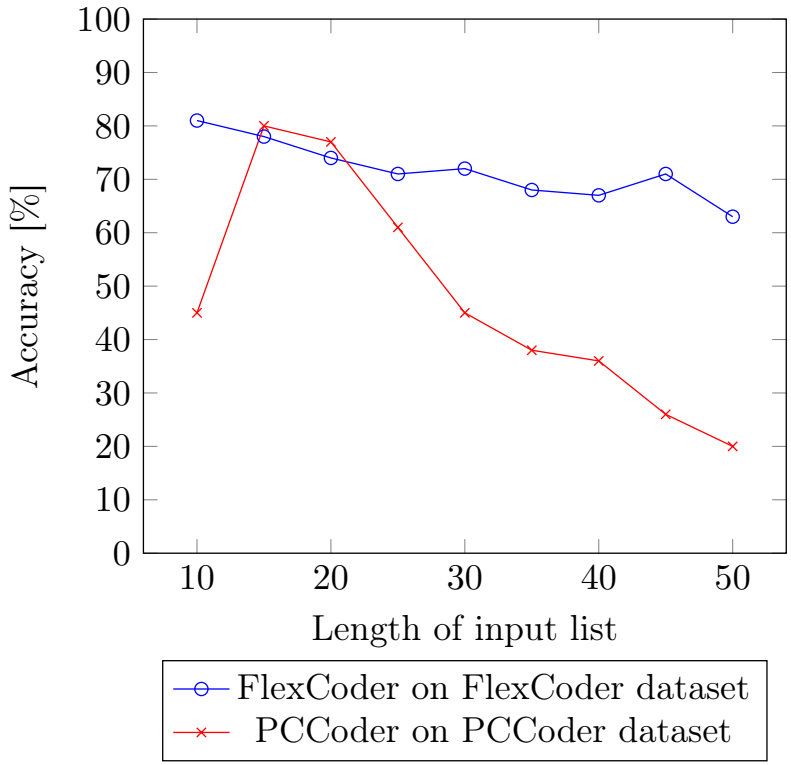
- The branch leading to success is indicated by black rectangles

Structure of Neural Network



Comparison With a State-of-the-Art System





- Our system: FlexCoder
- A Zohar, L Wolf: PCCoder

Takeaways

- We introduce a program synthesis system that generalizes well to different input lengths.
- We treat the operators in lambda functions separately from their parameters.
- We extend the range of intermediate results and outputs fourfold compared to previous works.

Thank you for your attention!

The materials were produced as part of EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.