Deep Learning for Program Synthesis from Diverse Specifications

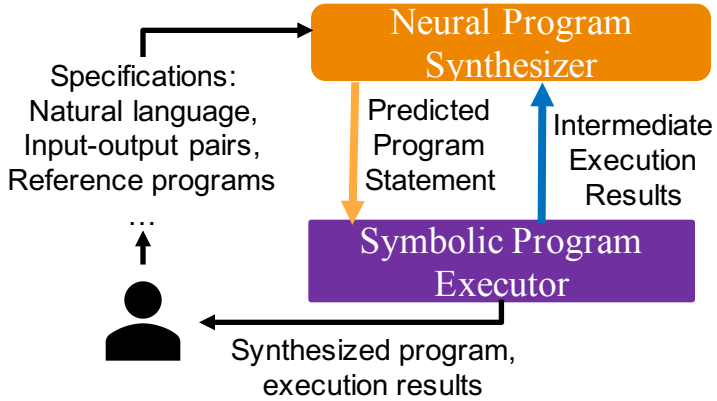
Xinyun Chen UC Berkeley

Motivation

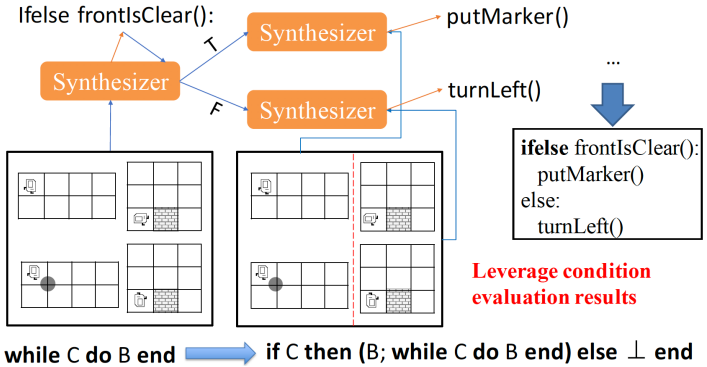
- **Neural program synthesis: deep learning** enables program synthesis from more **diverse specifications**.
- **Existing approaches** synthesize programs with **limited complexity**, and **may not generalize** to different inputs.
- **My works** design neural-symbolic models towards addressing these limitations, by leveraging **execution results**, **syntactic** and **semantic** properties of programs.

Neural-Symbolic Model

Our framework: **Neural Programs** operating a **Symbolic Execution Machine**.



Execution-Guided Synthesis: Beyond Synthesizing Sequential Programs



- Besides sequential programs, conditional statements and loops are also supported.
- SOTA on Karel benchmark, for program synthesis from input-output pairs.

Chen et al., Execution-Guided Neural Program Synthesis, ICLR 2019.

Neural-Symbolic Stack Machines: Compositional Generalization via Program Synthesis

- The symbolic stack supports recursion for execution.
- Categorizing execution traces reveal grammar rules.
- 100% accuracies on several compositional generalization benchmarks.

Chen et al., Compositional Generalization via Neural-Symbolic Stack Machines, NeurIPS 2020.
Chen et al., Towards Synthesizing Complex Programs from Input-Output Examples, ICLR 2018.