Syntax-Guided Program Synthesis

The syntax-guided program synthesis (SyGuS) problem aims to find a function $f$ that meets both syntactic and semantic constraints:

- **syntactic constraint:** a context-free grammar $G$ describing the syntax that captures the candidate implementations of $f$.
- **semantic constraint:** a logical formula $\phi$ that captures the desired functionality of $f$.

Cryptographic circuits synthesis

SYGUSSolver

Semantic Constraint $\forall (b_1, \ldots, b_n) \in (\{0, 1\})^n$

SyGuS Solver

Challenges

Representation learning and transfer learning of **structured data** with **rich semantics** remain open problems. Our work addresses two fundamental challenges in this area:

- How to learn a **neural** representation of both syntax and semantic constraints?
- How to learn a **transferable** policy for program synthesis tasks with **different** syntax $G$ and semantic $\phi$?

METAL: Meta-Learning Framework

Jointly learn the representation of syntax and semantics

Learn embedding via message passing

$$h^{t+1} = \text{aggregate}\left(\{h_t(e, w)\}\big|_{w \in \mathbb{W}}\right)$$

$$F(h^t, e) = \sigma(W^e h^t)$$

Graph construction objective:

- Capture the syntax structure and property (e.g., rule order invariance, symmetry)
- Capture the semantic constraint by its AST
- Enable information exchange by shared tokens

Learn embedding via message passing

Synthesize programs using a grammar adaptive policy network

Policy network is built in an auto-regressive way

$$\pi(f | \phi, G) = \prod_{t=1}^{|f|} \pi(a_t | \mathcal{H}(G), T^{(t-1)})$$

A2C is used for model training, reward is the passing ratio of accumulated examples.

$$\text{reward} = \sum_{b \in \mathbb{B}} \frac{|b \cap B^*|}{|b|}$$

$b$: counter example $B^*$: samples near $b$

Experimental Results

- Collected 214 circuit synthesis tasks from the 2017 SyGuS Competition
- Compared with one baseline solver and two state-of-the-art solvers
- Code and data are available at: https://github.com/PL-ML/metal

Learning to solve from scratch (overall: 214)

Visualization of solved tasks

Meta-learning enables faster solving (train/test: 150/64)