Learning a Meta-Solver for Syntax-Guided Program Synthesis

Xujie Si*, Yuan Yang*, Hanjun Dai, Mayur Naik, Le Song (*equal contribution)

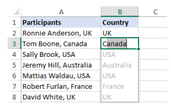
University of Pennsylvania & Georgia Institute of Technology

Syntax-Guided Program Synthesis

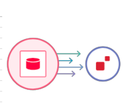
The syntax-guided program synthesis (SyGuS) problem aims to find a function f that meets both syntactic and semantic constraints:

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

Intelligent Tutor



End-user Programming

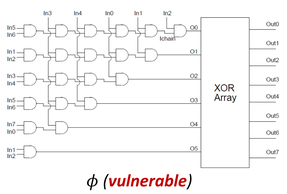


Data Transformation

Code Optimization

Code Auto-completion

Cryptographic circuits synthesis

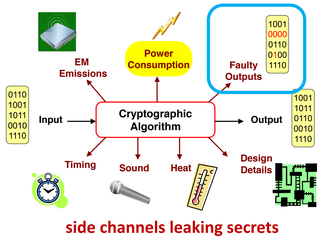


ϕ (***vulnerable***)

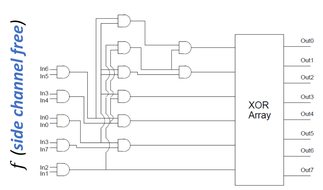
Grammar (Syntactic Constraint)
S -> d_1 XOR d_1 | d_1 AND d_1
d_1 -> d_2 AND d_2 | NOT d_2
d_2 -> ...
d_3 -> in0 | ... | in7

SyGuS Solver

Semantic Constraint
$\forall in_0, \ldots, in_7$
$\phi(in_0, \ldots, in_7) \equiv f(in_0, \ldots, in_7)$



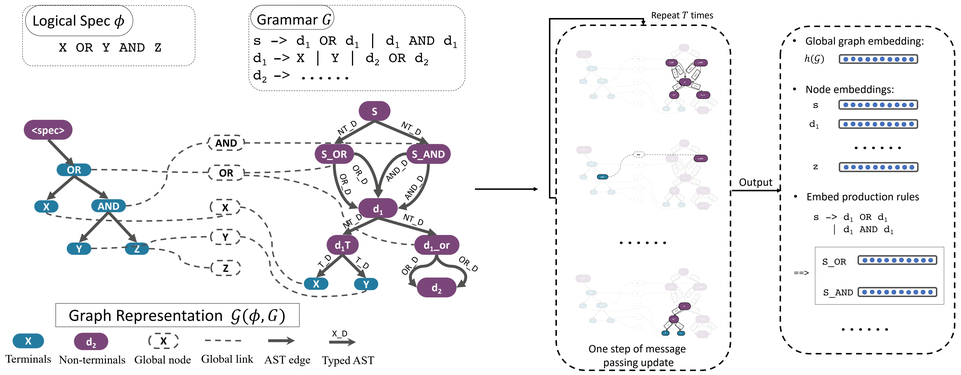
side channels leaking secrets



f (*side channel free*)

METAL: Meta-Learning Framework

Jointly learn the representation of syntax and semantics



Graph construction objective:

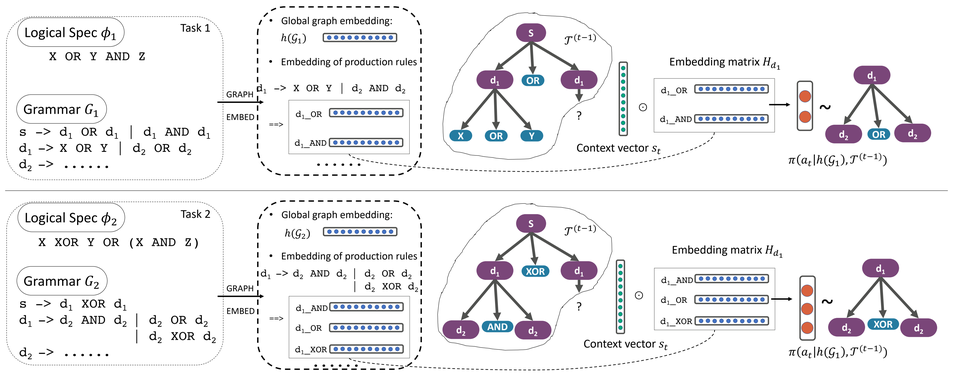
- Capture the syntax structure and property (e.g. rule order invariance, symmetry)
- Capture the semantic constraint by its AST
- Enable information exchange by shared tokens

Learn embedding via message passing

$$h_v^{t+1} = aggregate\left(\left\{F(h_u^t, e_{u,v})\right\}_{u \in N(v)}\right)$$

$$F(h^t, e) = \sigma(W_t^{e^\top} h^t)$$

Synthesize programs using a grammar adaptive policy network



Policy network is built in an auto-regressive way

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

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

$$reward = \frac{\sum_{b \in B_\phi \cup \widehat{B_b}} [f(b) \equiv \phi(b)]}{|B_\phi \cup \widehat{B_b}|}$$

b: counter example $\quad \widehat{B_b}$: samples near b

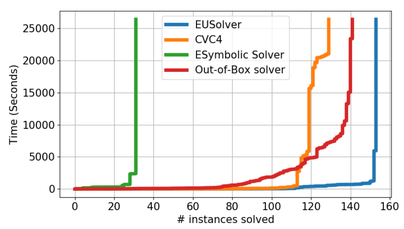
Challenges

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

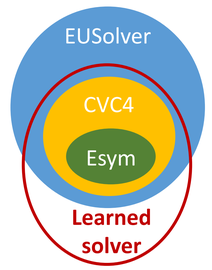
- ❑ How to learn a ***neural*** representation of both syntax and semantic constraints?
- ❑ How to learn a ***transferable*** policy for program synthesis tasks with ***different*** syntax ***G*** and semantic $\boldsymbol{\phi}$?

Experimental Results

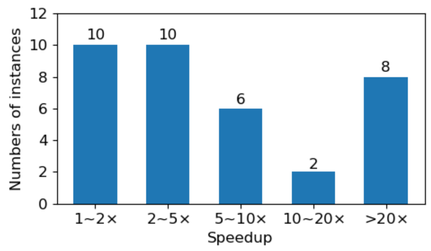
- Collected 214 circuit synthesis tasks from the 2017 SyGuS Competition
- Compared with one baseline solver and two state-of-the-art solvers
- Code and data are available at: https://github.com/PL-ML/metal



Learning to solve from scratch (overall: 214)



Visualization of solved tasks



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

Contact: {xsi,mhnaik}@cis.upenn.edu; {yyang754, hanjundai}@gatech.edu; lsong@cc.gatech.edu