The 5th Competition on Syntax-Guided Synthesis

Rajeev Alur, Dana Fisman, Rishabh Singh and Abhishek Udupa
Idea and Definition in a Nutshell
Program Synthesis

Specification $S$
High Level
“WHAT”

Synthesizer

Program $P$
Low Level
“HOW”
Recent Trends in Synthesis

Specification S
High Level
“WHAT”

Syntactic restrictions
R on the
solutions domain

Synthesizer

Program P
Low Level
“HOW”
Motivation:

- Tractability
- Combine human expert insights with computers exhaustiveness & rapidness
- Benefit progress SAT & SMT Solvers
SyGuS - The Vision

- Program Optimization
- Program Sketching
- Programming by examples
- Invariant Generation
- ?????

Program Optimization
Program Sketching
Programming by examples
Invariant Generation

SyGuS Syntax-Guided Synthesis
Syntax-Guided Synthesis (SyGuS) Problem

- Fix a background theory $T$: fixes types and operations

- Function to be synthesized: name $f$ along with its type
  - General case: multiple functions to be synthesized

- Inputs to SyGuS problem:
  - Specification $\varphi$
    - Typed formula using symbols in $T$ + symbol $f$
  - Context-free grammar $G$
    - Characterizing the set of allowed expressions $[[G]]$ (in theory $T$)

- Computational problem:
  - Find expression $e$ in $[[G]]$ such that $\varphi[f/e]$ is valid (in theory $T$)
SyGuS as Active Learning

**Concept class:** Set $[[G]]$ of expressions

**Examples:** Concrete input values
SyGuS-Comp18

The 5\textsuperscript{th} competition on Syntax Guided Synthesis
Solvers

- **CVC4 2018** - Andrew Reynolds (Univ. of Iowa), Haniel Barbosa (Univ. of Iowa), Andres Notzli (Stanford), Cesare Tinelli (Univ. of Iowa) and Clark Barrett (Stanford)

- **Horndini** - Deepak D'Souza (IISc Bangalore), P. Ezudheen (IISc Bangalore), P. Madhusudan - (UIUC), Pranav Garg (Amazon), Daniel Neider (MPI-SWS) and Shubham Ugare (IIT Guwahati)

- **DryadSynth** - KangJing Huang (Purdue Univ.), Xiaokang Qiu (Purdue Univ.), Qi Tian (Nanjing University), and Yanjun Wang (Purdue Univ.)

- **LoopInvGen** - Saswat Padhi (UCLA), Todd Millstein (UCLA) and Rahul Sharma (MSR)

- **EUSolver 2017** - Arjun Radhakrishna (MSR) and Abhishek Udupa (MSR)
Tracks

- General
- Inv
- CLIA
- PBE Strings
- PBE Bitvectors

Extensions suggestions?
Tracks Participation

- **CVC4-2018**: all 5 tracks
- **EUSolver-2017**: all 5 tracks
- **DryadSynth**: CLIA and INV tracks
- **LoopInvGen**: INV track
- **Horndini**: INV track
New Benchmarks

- **General (29)**  
  by Qinheping HU and Loris D’Antoni (Univ. of Wisconsin-Madison)

- **Invariant Generation (21+32)**  
  by Saswat Padhi (UCLA) + Kangjing Huang (Purdue Univ)

- **Conditional Linear arithmetic (15)**  
  by Kangjing Huang (Purdue Univ)

- **PBE Strings (10)**  
  by Woosul Lee (Penn)
Inv Track (127)

- LoopInvGen: 1st place

<table>
<thead>
<tr>
<th>Tool</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVC4 1.6</td>
<td>850</td>
</tr>
<tr>
<td>DryadSynth</td>
<td>784</td>
</tr>
<tr>
<td>LoopInvGen</td>
<td>966</td>
</tr>
<tr>
<td>Horndini</td>
<td>428</td>
</tr>
<tr>
<td>EUSolver</td>
<td>411</td>
</tr>
</tbody>
</table>
CLIA Track (88)

1st

DryadSynth & CVC4

Last year
CVC4
solved
73/73

Last year
DryadSynth
solved
32/73

Outstanding improvement!
PBE: Stings (118) Bitvectors (750)

CVC4

#1 last year

#4 last year
CVC4

Great improvement from last year (#2)

General Track (598)