FLoC Olympic Games 2014

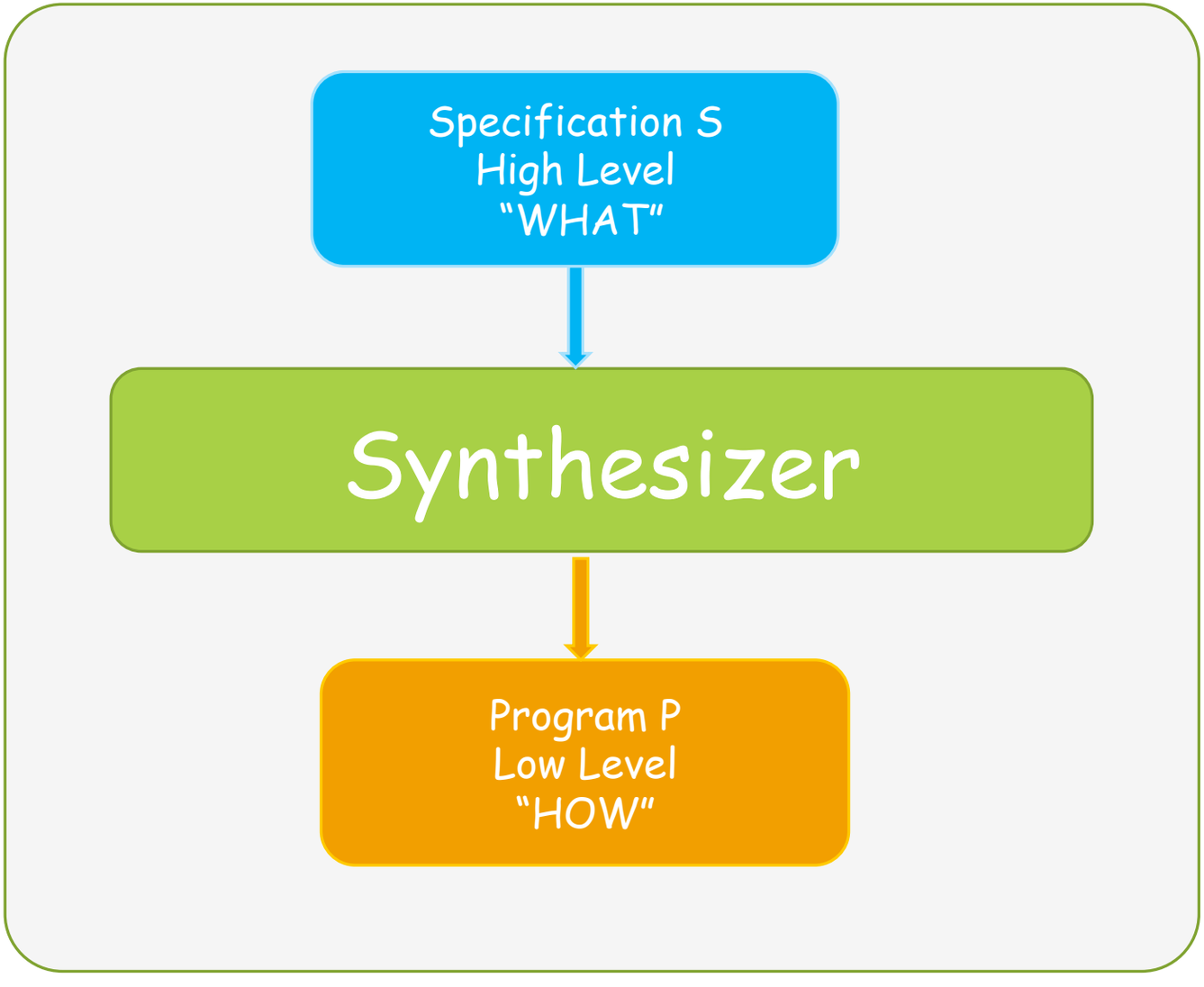
Citius, Maius, Potentius — Faster, Bigger, More Powerful

The First Syntax-Guided Synthesis Competition

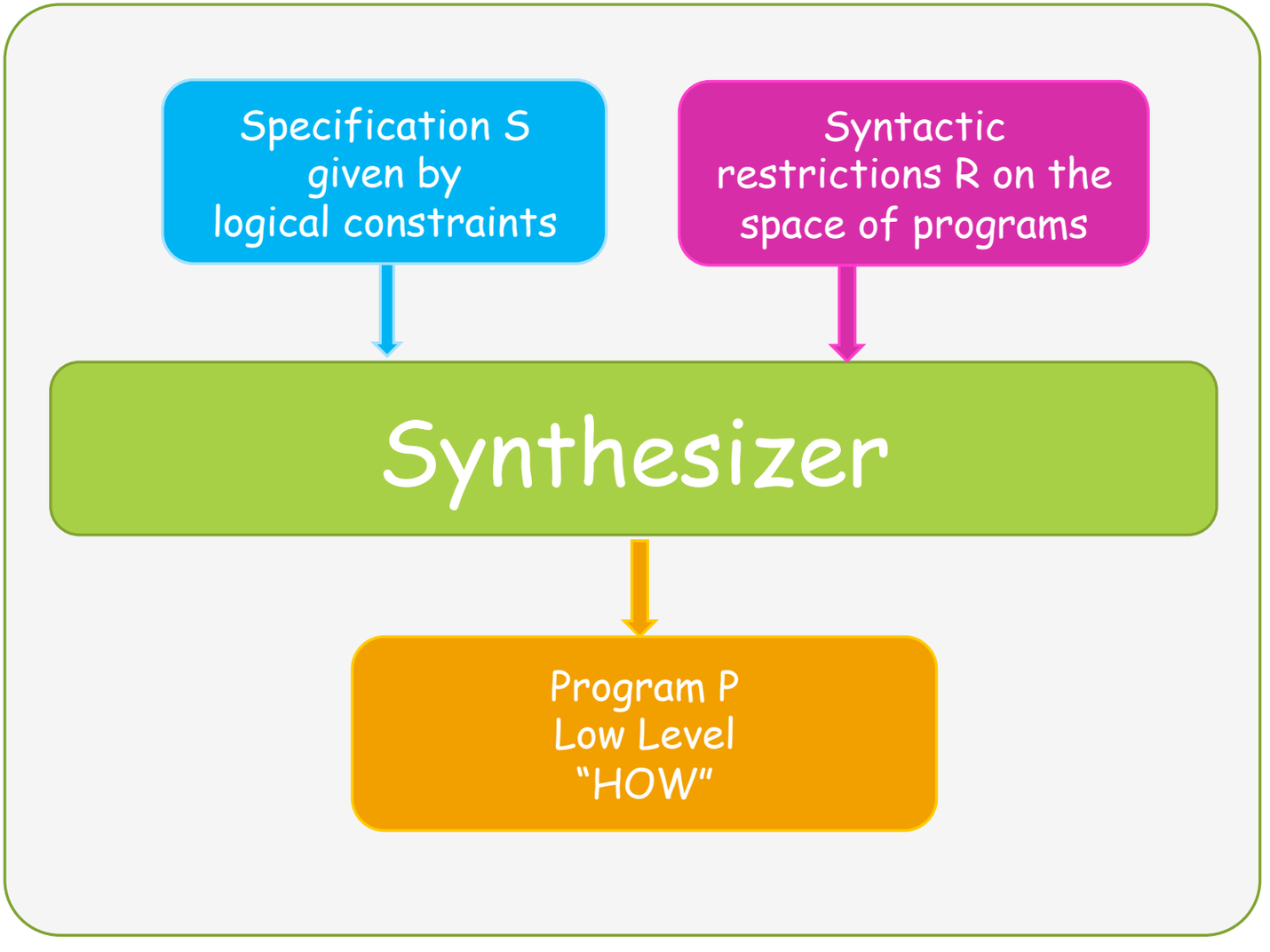
Organized by:

Rajeev Alur, Dana Fisman,
Rishabh Singh, Armando Solar-Lezama

Traditional Program Synthesis



Syntax-Guided Synthesis

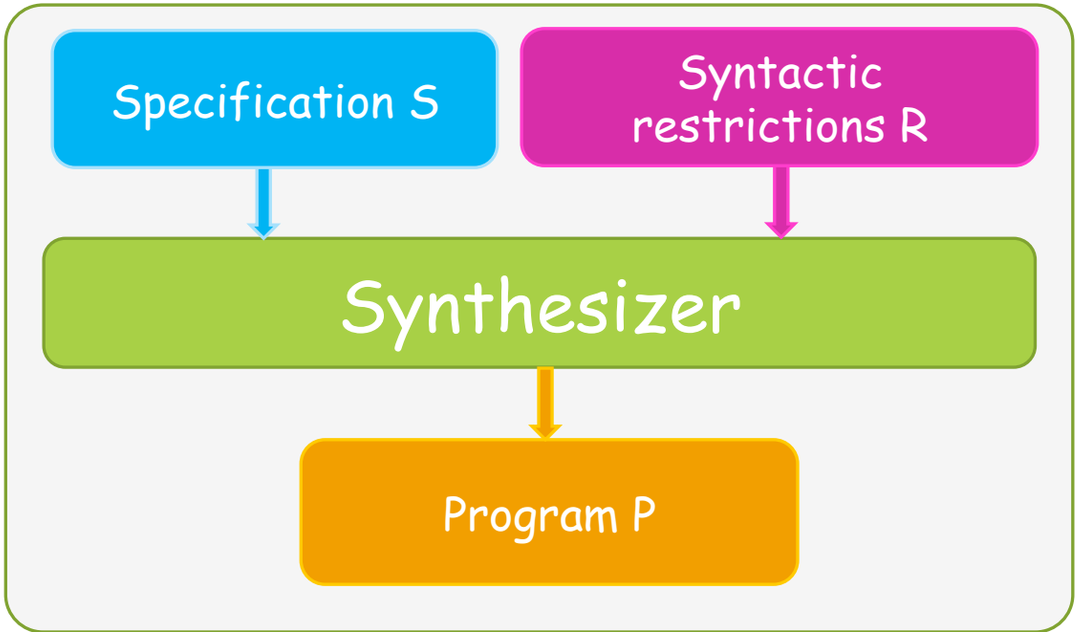


Syntax-Guided Synthesis

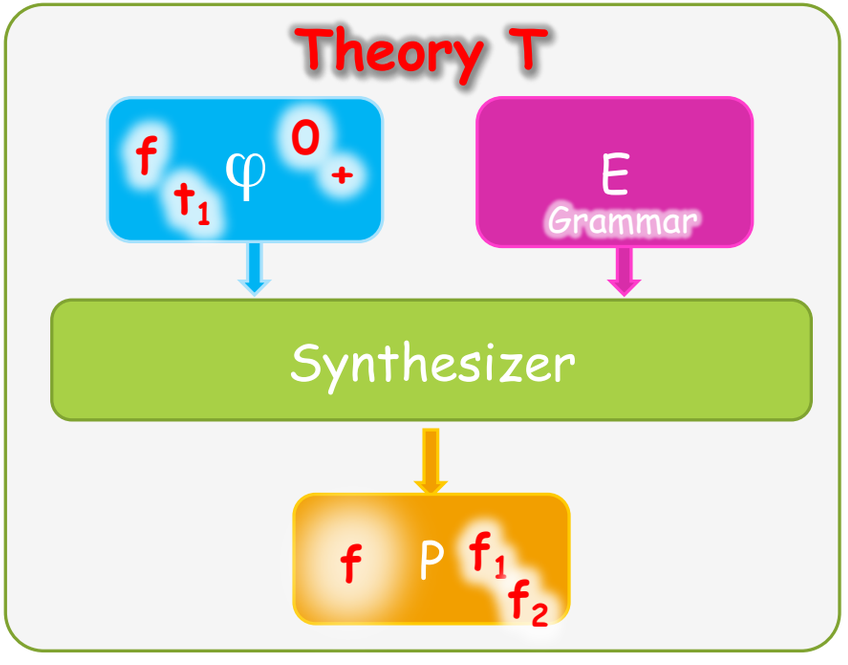
Common theme for many projects:

- Sketch (Bodik, Solar-Lezama et al)
- FlashFill (Gulwani et al)
- Super-optimization (Schkufza et al)
- Invariant generation (Many recent efforts…)
- TRANSIT for protocol synthesis (Udupa et al)
- Oracle-guided program synthesis (Jha et al)
- Implicit programming: Scala^Z3 (Kuncak et al)
- Auto-grader (Singh et al)

But no way to compare solutions and/or share benchmarks



Syntax-Guided Synthesis (SyGuS) Problem



- Fix a background theory T:
  fixes types and operations

- Function to be synthesized:
  name f (or f1, f2, …, fk) along with its type

- Inputs to SyGuS problem:
  - Specification φ

    Typed formula using symbols in T + symbol f
  - Context-free grammar G

    Characterizing the set of allowed expressions E (in theory T)

- **Computational problem:**

  Find expression e in E such that φ[f/e] is valid (in theory T)

SyGuS - example

- Theory QF-LIA
  Types: Integers and Booleans
  Logical connectives, Conditionals, and Linear arithmetic
  Quantifier-free formulas

- Function to be synthesized f (int x, int y) : int

- Specification: (x ≤ f(x,y)) & (y ≤ f(x,y)) & (f(x,y)=x | f(x,y)=y)

- Candidate Implementations: Linear expressions
  LinExp := x | y | Const | LinExp + LinExp | LinExp - LinExp

- No solution exists

SyGuS - example

- Theory QF-LIA

  Types: Integers and Booleans

  Logical connectives, Conditionals, and Linear arithmetic

  Quantifier-free formulas

- Function to be synthesized f (int x, int y) : int

- Specification: (x ≤ f(x,y)) & (y ≤ f(x,y)) & (f(x,y)=x | f(x,y)=y)

- Candidate Implementations: Conditional expressions with comparisons

  Term := x | y | Const | If-Then-Else (Cond, Term, Term)

  Cond := Term <= Term | Cond & Cond | ~Cond | (Cond)

- Possible solution:

  If-Then-Else (x ≤ y, y, x)

Benchmark Problems

- Hacker's Delight (bit manipulation problems)
- Invariant Generation (for program verification) [Garg et al.]
- Vehicle Control (autonomous cars on shared routes) [Dallal et al.]
- Conditional integer arithmetic (complex branching structure)
- ICFP (bit vector algorithms from functional programming competition) [Swamy et. al.]
- And others…

SyGuS
Syntax-Guided Synthesis

Solvers

In this year's competition participated 5 solvers:

- Enumerative CEGIS Solver (Abhishek Udupa, Penn)
- Stochastic CEGIS Solver (Mukund Raghothaman, Penn)
- Symbolic CEGIS Solver (Garvit Juniwal, UC Berkeley)
- Alchemist (Pranav Garg, Shambwaditya Saha, P. Madhusudan, UIUC)
- Sketch-Based (Rishabh Singh, Armando Solar-Lezama, MIT)

And the Winner is:

Enumerative-CEGIS Solver

Abhishek Udupa, Penn.

Thanks !

- StarExec for providing computational infrastructure
- Input format extends SMTLib-2
- Microsoft Research: Sponsorship of financial award
- NSF Expeditions project ExCAPE and its team members

ExCAPE
Expeditions in Computer Augmented Program Engineering

SyGuS
Syntax-Guided Synthesis