Expeditions in Computer Augmented Program Engineering

http://excape.cis.upenn.edu/

Cornell, Maryland, Michigan, MIT, Penn, Rice, UC Berkeley, UCLA, UIUC

Annual PI Meeting, May 2016
ExCAPE Vision
Synthesis for
Programming Assistance

- Designer expresses “what”, using multiple input formats
- Synthesizer discovers new artifacts via integration
- Synthesizer solves computationally demanding problems using advanced analysis tools
- Interactive iterative design
- Integrated formal verification
## Research Organization

<table>
<thead>
<tr>
<th>Tools and Evaluation</th>
<th>Design Methodology</th>
<th>Computational Engines</th>
<th>Challenge Problems</th>
<th>Education and Knowledge Transfer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Programming for Mobile Platforms</td>
<td>Multicore Protocols</td>
</tr>
</tbody>
</table>
Talk Outline

- Syntax-guided synthesis
- Education, collaborations, knowledge transfer, and outreach
- Emerging and future directions
Classical Program Synthesis

Specification $S$
High Level
"WHAT"

Synthesizer

Program $P$
Low Level
"HOW"
Syntax-Guided Synthesis

Specification $S$ given by logical constraints

Syntactic restrictions $R$ on the space of programs

Synthesizer

Program $P$
Application: Superoptimizing Compiler

- Given a program $P$, find a “better” equivalent program $P'$

```c
#define shift(x, d) (x << d)
multiply (x[1,n], y[1,n]) {
    x1 = x[1,n/2];
    x2 = x[n/2+1, n];
    y1 = y[1, n/2];
    y2 = y[n/2+1, n];
    a = x1 * y1;
    b = shift( x1 * y2, n/2);
    c = shift( x2 * y1, n/2);
    d = shift( x2 * y2, n);
    return ( a + b + c + d)
}
```

Replace with equivalent code with only 3 multiplications
Application: Programming by Examples

- Find a program that is consistent with given input/output examples

- String transformation program:
  - “Benjamin C. Pierce” $\rightarrow$ “B. Pierce”
  - “Lee, Insup” $\rightarrow$ “I. Lee”
  - “Patrice Godefroid” $\rightarrow$ “P. Godefroid”

- Syntax specifies what transformations are permissible
  - Conditionals not allowed (rules out trivial if-then-else programs)
  - Pattern matching using regular expressions allowed

- Success in Microsoft’s FlashFill system for Excel
Syntax-Guided Program Synthesis

- Core computational problem: Find a program $P$ such that
  1. $P$ is in a set $E$ of programs (syntactic constraint)
  2. $P$ satisfies $\text{spec } \varphi$ (semantic constraint)

- Common theme to many recent efforts
  - Sketch (Bodik, Solar-Lezama et al)
  - FlashFill (Gulwani et al)
  - Super-optimization (Schkufza et al)
  - Invariant generation (Many recent efforts...)
  - Genetic programming + model checking (Peled et al)
  - TRANSIT for protocol synthesis (Udupa et al)
  - Oracle-guided program synthesis (Jha et al)
  - Implicit programming: Scala$^\text{Z3}$ (Kuncak et al)
  - Auto-grader (Singh et al)
Inspiration: SMT Success Story

SMT-LIB Standardized Interchange Format (smt-lib.org)
Problem classification + Benchmark repositories
LIA, LIA_UF, LRA, QF_LIA, ...

+ Annual Competition (smt-competition.org)

- CBMC
- SAGE
- VCC
- Spec#

- Z3
- Yices
- CVC4
- MathSAT5
SyGuS Solvers ↔ Synthesis Tools

- Program optimization
- Program sketching
- Programming by examples
- Invariant generation

SYNTH-LIB Standardized Interchange Format
Problem classification + Benchmark repository

+ SyGuS-COMP (Competition for solvers)

Potential Techniques for Solvers:
Learning, Constraint solvers, Enumerative/stochastic search
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
  - Set E of expressions given by a context-free grammar
    - Set of candidate expressions that use symbols in T

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

Initial examples I

Learning Algorithm

Verification Oracle

Candidate Expression

Counterexample

Fail

Success

Concept class: Set E of expressions

Examples: Concrete input values
SyGuS Benchmarks

- Over 500 benchmarks (see www.sygus.org)
- Hacker’s Delight: Tricky bit-vector manipulation programs
- Invariant generation: From software verification competition
- Robotic controller: Autonomous vehicle routing
- ICFP Programming competition
- FlashFill string transformation benchmarks (new in 2016!)
SyGuS 2014

- Solvers based on CEGIS (Counter-example guided inductive synthesis) approach

- Learning strategies based on:
  - Enumerative (search with pruning): Udupa et al (PLDI’13)
  - Symbolic (solving constraints): Gulwani et al (PLDI’11)
  - Stochastic (probabilistic walk): Schkufza et al (ASPLOS’13)

- Competition of solvers held as part of FLoC

- Winner: Enumerative solver (Udupa)
SyGuS 2015

- New computational approaches proposed
  - Multiple papers in CAV / POPL / ...

- New users
  - Program optimization

- New Features in 2015 Competition
  - Special track for conditional linear integer arithmetic (fixed syntax)
  - Special track for invariant synthesis (fixed specification template)

- New entries: Alchemist, CVC4, ICE, SosyToast

- Winners:
  - CVC4 in general track and conditional linear arithmetic
  - ICE in invariant synthesis
SyGuS 2016

- New computational approaches are emerging
  - Divide and conquer search strategy based on decision trees
  - Type theoretic tools + search
  - ICFP benchmarks solved for the first time

- Competition will be held at CAV (in July)
  - New special track for programming by examples
  - Expect a couple of new of entries for solvers

- Increasing interest in formal methods community
  - Tutorial at POPL 2016
  - Lectures at Marktoberdorf summer school
  - Special issue of “Formal Methods in System Design”
  - Over 100 citations for SyGuS paper within 2 years of publication
Collaborations, Education, Knowledge Transfer, and Outreach
COLLABORATION CHALLENGE

9 Universities:
Cornell, Maryland, Michigan, MIT, Penn, Rice, UC Berkeley, UCLA, UIUC

18 Principal Investigators

8 Disciplines:
COLLABORATIONS ACROSS DISCIPLINES
COLLABORATIONS ACROSS INSTITUTIONS
Rotating Postdoc Program

Each ExCAPE postdoc has two mentors, at two different institutions

1. Ruediger Ehlers (2012-13), Kress-Gazit and Seshia
2. Christos Stergiou (2013-15), Alur and Tripakis
3. Xiaokang Qiu (2013-15), Foster and Solar-Lezama
4. Indranil Saha (2013-15), Pappas and Seshia
5. Daniel Neider (2014-16), Parthasarathy and Tabuada
7. Gustavo Soares (2016), Bodik and Hartmann
Collaboration with Industry

- **Industrial Advisory Board**
  - Brunell (GE Research), Fix (Intel), Chandra (Samsung),
  - Godbole (Honeywell), Godefroid (Microsoft), Gupta (NEC),
  - Kuehlmann (Coverity), Mosterman (Mathworks),
  - Wegman (IBM), Zave (AT&T)

- **Research collaborations with industry researchers**
  - HP Labs, Intel, Microsoft, Samsung, Mozilla, GreenArrays, Toyota

- **SRC/DARPA Research Center TERRASWARM**

- **Graphistry**: Startup founded by ExCAPE alumni

- **NSF ICORPS program (NetEgg project @ Penn)**
Knowledge Transfer

- Summer school 2013 and 2015
- Annual Workshop: SYNT
- Special sessions, tutorials, and affiliated workshops in many conferences and summer schools (e.g. Dagstuhl, Marktoberdorf)
- Monthly Webinar: starting Sept 2012, 40 talks (publicly available)
ExCAPE Summer School

- **First edition:** June 12 - 15, 2013 at UC Berkeley
  - Three 3-hour tutorials + “hands-on” sessions
  - Invited talks
  - Accommodation and registration and food funded by ExCAPE
  - 90 participants from 12 different countries
  - Highly positive feedback in post-school surveys

- **Second edition:** June 23 - 26, 2015 at MIT
  - 85 participants
  - 10 women
  - 18 participants from outside US: Austria, Brazil, England, Finland, Germany, Iran, Italy, Japan, Kazakhstan, and Switzerland
  - Program consisting of tutorials and invited talks
  - Free registration and food
Education Technology

- Tutoring technology based on constraint solvers and synthesis tools
  - Problem generation
  - Grading
  - Feedback generation

- Current tools
  - AutomataTutor
  - AutoProf
  - CPSGrader
  - Data Science course (in progress)

- Already in use in classrooms and MOOCs

ExCAPE Alumni I (2013—16)

- Sina Caliskan (PhD, UCLA -> Mathworks)
- Alvin Cheung (PhD, MIT -> Faculty, U. Washington)
- Eric Dallal (PhD, Michigan -> Postdoc, UCLA)
- Loris D’Antoni (PhD, Penn -> Faculty, U. Wisconsin, Madison)
- Ruediger Ehlers (ExCAPE postdoc -> Faculty, U. Brmen, Germany)
- Pranav Garg (PhD, UIUC -> Amazon)
- Alex Gurney (Postdoc, Penn -> Comcast)
- Thibaud Hottelier (PhD, UC Berkeley -> Graphistry)
- Sela Mador-Haim (ABD, Penn -> Coverity)
- Jinseong Jeon (PhD, Maryland, -> Google)
- Leo Meyerovich (PhD, UC Berkeley -> Graphistry)
ExCAPE Alumni II (2013—16)

- Peter-Michael Osera (PhD, Penn -> Faculty, Grinnell College)
- Xiaokang Qiu (ExCAPE postdoc -> Faculty, Purdue)
- Arun Raghavan (PhD, Penn -> Oracle Labs)
- Vasumathi Raman (PhD, Cornell -> Postdoc, CalTech)
- Matthias Rungger (PhD, UCLA -> Postdoc, TU Munich)
- Indranil Saha (ExCAPE Postdoc -> Faculty, IIT Kanpur, India)
- Rishabh Singh (PhD, MIT -> Microsoft Research)
- Christos Stergiou (ExCAPE Postdoc -> Google)
- Yasser Shoukry (PhD, UCLA -> Postdoc, UCLA/UC Berkeley)
- Emina Torlak (Research scientist, UC Berkeley -> Faculty, U. Washington)
- Abhishek Udupa (PhD, Penn -> Microsoft)
- Anduo Wang (PhD, Penn -> Postdoc, UIUC -> Faculty, Temple U)
- Yi-Chin Wu (PhD, Michigan -> Postdoc, Michigan/UC Berkeley)
New Directions
- ExCAPE Faculty: Alur, Bodik, Hartmann, Seshia, Solar-Lezama, Vardi
- ExCAPE Alumni: D’Antoni, Singh
- Researchers from many disciplines, particularly, cognitive learning science
How to specify quantitative policies over data streams?

Synthesis of optimized streaming implementation from high-level declarative policy specification

ExCAPE research results:
Foundation of quantitative regular expressions (QRE)
Toolkit NetQRE for networked systems
Beyond Autonomous Control

Driverless Cars  Coordinating robots  Medical devices

ExCAPE theme:
how to synthesize controller from correctness requirements?

Beyond ExCAPE:
- Accounting for human-in-the-loop
  NSF CPS Frontiers (to be announced), PI: Seshia
- Integrating autonomous components in existing infrastructure
Scaling up Synthesis

- Synthesis to assist programmers in using components, libraries ...
  - Formal methods
    + Software engineering
    + Machine learning

- Starting points:
  - ExCAPE projects CodeHint, Pasket
  - DARPA MUSE program