Expeditions in Computer Augmented Program Engineering

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

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

Annual PI Meeting, June 2015
Formal Methods: State of the art

- Powerful verification tools
  - SAT/SMT solvers
  - Model checkers
  - Proof assistants

- Applications to real-world problems
  - Hardware verification
  - Software testing for security bugs
  - Safety-critical control software

- Limitations
  - Scalability...
  - Used by verification engineers, but not by programmers

Can we leverage formal analysis to transform system design process?
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

Design Methodology

Computational Engines

Challenge Problems

Programming for Mobile Platforms
Multicore Protocols
Networked Systems
Robotic Systems
Personalized Education

Education and Knowledge Transfer

Tools and Evaluation
Talk Outline

- Syntax-guided synthesis
- AutomataTutor
- Education, collaborations, knowledge transfer, and outreach
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
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
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 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^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
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

Candidate Expression

Verification Oracle

Counterexample

Fail

Success

Concept class: Set E of expressions

Examples: Concrete input values
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

Little engines of synthesis?
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
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)
New computational approaches are emerging
- Multiple submissions to CAV 2015 on solving SyGuS
- Talks today and tomorrow

New users
- Program optimization

Competition will be held at CAV (in July)
- Special track for conditional linear integer arithmetic (fixed syntax)
- Special track for invariant synthesis (fixed specification template)
- Expected entries from wider CAV community (e.g. CVC4)
Automata Tutor and what we learned from building an online teaching tool

Rajeev Alur (Penn), Loris D’Antoni (Penn), Sumit Gulwani (MSR), Bjoern Hartmann (Berkeley), Dileep Kini (UIUC), Mahesh Viswanathan (UIUC), Matt Weaver (Penn), Alex Weinert (RWTH)
What is AutomataTutor?

Or NFA, or regular expression

Draw the DFA accepting the language:

\{ s \mid \text{‘}ab\text{‘} \text{ appears in } s \text{ exactly 2 times } \}

Solution:
The problem description was

\[
\{ \ s \mid \text{‘}ab\text{’ appears in} \ s \ \text{exactly} \ 2 \ \text{times} \ \}\]

The student instead drew the DFA that accepts

\[
\{ \ s \mid \text{‘}ab\text{’ appears in} \ s \ \text{at least} \ 2 \ \text{times} \ \}\]

**INTUITION:** find the edit distance between the two language descriptions
Feedback via Synthesis

\[ \text{indOf}(ab) = 2 \]

\[ \text{indOf}(ab) \geq 2 \]

Replace \( \geq \) with \( = \)

\{ s \mid 'ab' appears in s at least 2 times \}
Tool evaluation

• User studies
  – Tool and humans grade similarly [IJCAI13]
  – Student receiving feedback solve more practice problems [TOCHI13]

• Surveys
  – Students like the feedback and find it helpful
  – The drawing interface is easy to use

• Instructors’ feedback
  – Students are more engaged and perform better
Instructors enjoy the tool

“This is how the construction of finite automata that recognize regular languages should be taught in a modern way! I wish I had similar tools for all the topics I need to cover.”

--- Luca Aceto, Ode to Automata Tutor
What did we learn?

What worked

• Good interfaces are important
• Simple feedback is good enough
• Instructors like independence
• Instructors love automated grading
• End-of-course surveys are extremely helpful

What did not work

• Verbose feedback is confusing
  – E.g. Your solution is incorrect on all strings that start with an ‘a’, end with a ‘b’, and do not contain a ‘c’
• How-to feedback works poorly if the student solutions is too far off
  – E.g. To fix your DFA you need to change the acceptance condition of 3 states and redirect 6 transitions…
• A single crash can cost many users
13 Universities using it
UIUC, Penn, UC San Diego, Cambridge, Reykjavik University, University of Liverpool, EPFL, Mahidol University International College, University of Louisiana at Monroe, IIIT Delhi, John Hopkins, Federal University of Campina Grande, University of Luxembourg

>3000 students
>50000 assignments graded
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
FOSTERING NEW COLLABORATIONS - EXAMPLE: ROBOTICS WORKSHOP AT RICE

Organization: Kavraki (Rice), Kress-Gazit (Cornell) & Vardi (Rice)

Attending groups: Cornell, Michigan, Penn, Rice, Berkeley and UCLA.

28 attendees (20 directly affiliated with ExCAPE, 8 from Rice groups)

Workshop goals and outcome:

1. Hands-on tutorial on synthesis tools for robotic systems
   - Open Motion Planning Library (OMPL) - Kavraki’s group
   - Linear Temporal Logic MissiOn Planning (LTLMoP) - Kress-Gazit’s group
   - Pessoa - a tool for embedded software synthesis - Tabuada’s group

2. Project updates through presentations

3. Discussions on possible future directions and collaborations.
   - Comparison of different synthesis approaches to robotic systems (Cornell, UCLA, Rice)
   - Design of controllers for continuous systems from LTL specifications (UCLA, Rice)
   - Solving invariant sets through SAT based techniques (UCLA, Berkeley)
   - Sampling-based construction of LQR funnels (UCLA, Rice)
   - Robotics benchmarks for Syntax-Guided Synthesis (Cornell, Michigan, Berkeley)
   - Quantitative temporal logic for reasoning about robustness (Penn, UCLA, Rice)
Collaboration - Example: Syntax Guided Synthesis

Summer 2012, ExCAPE meeting: Discussion of common format started

Spring 2013, Berkeley PI meeting: Concrete plans for SyGuS formalization

Telecons: Groups from Berkeley, MIT, Penn collaborate to
  - define syntax,
  - implement solvers,
  - collect benchmarks

Oct 2013: SyGuS paper presented at FMCAD

Dec 2013: Competition announced,
  SyGuS-Community builds
  - Mailing List, Website,
  - StarExec (collab with NSF Infrastructure project at U. Iowa)

Spring 2014, Robotics workshop: Groups from Michigan, Cornell initiate work on robotics benchmarks

Summer 2014: Competition at FLOC 2014.
  - Group from UIUC contribute benchmarks and solver.

Summer 2014: Discussion between Berkeley, MIT, Rice on using sampling to solve SyGuS

July 2014: SyGuS lectures at Marktoberdrof summer school

Nov 2014: Joint paper submitted to NATO proceedings
  - Groups from Berkeley, Cornell, Michigan, MIT, Penn, UIUC
Rotating Postdoc Program

- Each ExCAPE postdoc has two mentors, at two different institutions

- Ruediger Ehlers (2012-13), Kress-Gazit and Seshia
- Christos Stergiou (2013-15), Alur and Tripakis
- Xiaokang Qiu (2013-15), Foster and Solar-Lezama
- Indranil Saha (2013-15), Pappas and Seshia
- Daniel Neider (2014-16), Parthasarathy and Tabuada
- Eunsuk Kang (2015-16), Lafortune and Tripakis
- Gustavo Soares (2015-16), Bodik and Hartmann
ExCAPE Outreach: Tools

Infrastructure

- Rosette
- Sketch
- SyGuS
- OMPL

Domain specific

- AutomataTutor (Automata problems)
- AutoProf (Programming problems)
- CodeHint (API usage)
- ComPlan (Robotic motion planning)
- CPSGrader (CPS model-based testing)
- DReX (String transformations)
- Gadera (Controller synthesis)
- LTLMoP (Reactive synthesis)
- NetEgg (Network policies)
- Pasket (Patterns in event-driven apps)
- Pessoa (Robust synthesis)
- QBS (Database queries)
- Transit (Distributed protocols)
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**

- **Student internships**

- **Graphistry: Startup founded by ExCAPE alumni**
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, 30 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

- 80 participants
- 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

- Already in use in classrooms and MOOCs


Integrated with research on Personalized Education Theme !!
ExCAPE Alumni (2013–15)

- 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)
- Pranav Garg (PhD, UIUC -> Amazon)
- Alex Gurney (Postdoc, Penn -> Comcast)
- Ruediger Ehlers (ExCAPE Postdoc -> Faculty, U. Bremen, Germany)
- Thibaud Hottelier (PhD, UC Berkeley -> Graphistry)
- Sela Mador-Haim (MS, Penn -> Coverity)
- Leo Meyerovich (PhD, UC Berkeley -> Graphistry)
- Peter-Michael Osera (PhD, Penn -> Faculty, Grinnell College)
- Arun Raghavan (PhD, Penn -> Oracle Labs)
- Vasu 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)
- Anduo Wang (PhD, Penn -> Postdoc, UIUC -> Faculty, Temple U)
- Yi-Chin Wu (PhD, Michigan -> Postdoc, Michigan/UC Berkeley)