Education and Outreach

ExCAPE Kick-off Meeting
Monday June 4, 4:30-5:30pm

Leads: Stéphane Lafortune and Steve Zdancewic
Education and Outreach

• 2013 Summer School
• Programs for High School Students
• Intelligent Tutoring Software
• Course Modules

• Others (from proposal): Annual workshop; integrative training, undergraduate education, diversity enhancement
Education and Outreach

• *2013 Summer School*
• Programs for High School Students
• Intelligent Tutoring Software
• Course Modules

• Others (from proposal): Annual workshop; integrative training, undergraduate education, diversity enhancement
Planning the 2013 Summer School

- Proposal: Summer Schools in 2013, 2015, 2017
- Location? Date? Duration?
- Target audience (beyond ExCAPE team)
- Format and expected size
- Local organizers (volunteers welcome)

- Prior experiences
- Decision time frame
Education and Outreach

• 2013 Summer School
• Programs for High School Students
• Intelligent Tutoring Software
• Course Modules

• Others (from proposal): Annual workshop; integrative training, undergraduate education, diversity enhancement
Outreach to High School Students

• Several programs at participating institutions
  – Cornell
  – UPenn
  – Michigan
• “Programming is not coding”
• Several PIs have relevant experience
  – Hadas Kress-Gazit
  – others
Outreach to High School Students - Cornell

• CURIE academy – week long summer camp for high school girls
  – HKG ran the project component in 2010 and will do so again in 2012 and 2014
Outreach to High School Students - Cornell

• Expanding Your Horizons (EYH) – Day of workshops for middle school girls
  – Workshop using the iRobot Create – program using the bump sensor
UPenn Outreach programs

• Summer Academy in Applied Science & Tech.
  – Strong connections to Penn CIS Faculty

• Women in Computer Science
  – High School Day for Girls

• Dining Philosophers/Penn Apps.
  – Hackathons
Relevant Programs at Other Institutions

• Michigan Center for Engineering Diversity and Outreach
  – Spring courses with Detroit Area Pre-College Engineering Program (7th-9th graders)
  – Summer Engineering Academy (middle & high school)
• Others (discussion)
Education and Outreach

• 2013 Summer School
• Programs for High School Students
• Intelligent Tutoring Software
• Course Modules

• Others (from proposal): Annual workshop; integrative training, undergraduate education, diversity enhancement
Intelligent Tutoring Software

• A component of our “knowledge transfer”

• Target audience: High school and undergraduate students

• Scope of such tools
  – Two examples
Synthesis in Education

Armando Solar-Lezama
with slides from Rishabh Singh and Sumit Gulwani
The real software problem

• The Software Quality problem is a symptom

• The real problem:
  The demand for software in our society far exceeds
  the supply of people skilled enough to produce it
A new agenda

• Apply the lessons learned to:
  – Make programmers more productive
  – Make programming more accessible
  – Reduce the cost of training the next generation
Grading Programming Assignments

• Test-cases based grading
  – No precise correctness correlation
  – No student tailored feedback

• Manual grading by TAs
  – Error-prone, time consuming
Buggy Program for Array Reverse

using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for(int i=a.Length; i < a.Length; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}

Buggy Program for Array Reverse

```csharp
using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length - 1; i > a.Length - 2; i--) {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```
Buggy Program for Array Reverse

```csharp
using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length - 1; i < a.Length - 1; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```

No change! Sign of Frustation?
Buggy Program for Array Reverse

using System;
public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for(int i=a.Length; i <= a.Length; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}

20
public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length; i < a.Length; i--)
        {
            Console.WriteLine(i);
            b[count] = a[i];
            count++;
        }
        return b;
    }
}

Same as initial attempt except Console.WriteLine!
Buggy Program for Array Reverse

using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length; i < a.Length; i--)
        {
            Console.WriteLine(i);
            b[count] = a[i];
            count++;
        }
        return b;
    }
}

No change! Sign of Frustation?
Buggy Program for Array Reverse

using System;
public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for(int i=a.Length; i <= a.Length; i--)
        {
            Console.WriteLine(i);
            b[count] = a[i];
            count++;
        }
        return b;
    }
}

The student has tried this before!
Buggy Program for Array Reverse

```csharp
using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for(int i=a.Length; i < a.Length; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```

Same as initial attempt!
using System;

public class Program {

    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length - 1; i < a.Length - 1; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}

The student has tried this before!
Buggy Program for Array Reverse

```csharp
using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length; i > 0; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```

Almost correct! (a[i-1] instead of a[i] in loop body)
Buggy Program for Array Reverse

```csharp
using System;
public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length; i >= 0; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```

Student going in wrong direction!
Buggy Program for Array Reverse

```csharp
using System;
public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length; i < a.Length; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```

Back to bigger error!
Buggy Program for Array Reverse

```csharp
using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length; i < a.Length; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```

No change! Fruståtion!
Buggy Program for Array Reverse

```csharp
using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length - 1; i >= 0; i--) {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}
```

No change! More Frustation!!
Buggy Program for Array Reverse

using System;

public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        int count = 0;
        for (int i = a.Length; i < a.Length; i--)
        {
            b[count] = a[i];
            count++;
        }
        return b;
    }
}

No change! Too frustrated now!!! Gives up.
AutoGrader

• Automate grading
  – Find semantic errors
  – Feedback to fix them

Students make similar mistakes
using System;
public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        for (int i = 0; i < a.Length; i++)
        {
            b[a.Length-i] = a[i-1];
        }
        return b;
    }
}

Challenge 1: Different Algorithms

```csharp
using System;
public class Program {
    public static int[] Puzzle(int[] a) {
        int[] b = new int[a.Length];
        for (int i = 0; i < a.Length; i++)
            b[a.Length - 1 - i] = a[i - 1];
        return b;
    }
}
```

```csharp
using System;
public class Program {
    public static int[] Puzzle(int[] a) {
        int front, back, temp;
        front = 0;
        back = a.Length - 1;
        temp = a[back];
        while (front > back)
            { a[back] = a[front];
              a[front] = temp;
              ++back;
              ++front;
              temp = a[back];
            }
        return a;
    }
}
```
Challenge 2: Scalability

$10^{10}$ different possible candidate corrections
Our Approach

Use data of previous student solutions

Correction rules based on corrections

\[ a < b \rightarrow a \{<= | > | >= | == | !=\}b \]
\[ a[i] \rightarrow a[\{i + 1|i - 1|i-?i\}] \]

Create a set of candidate solutions using rules and find closest correct solution.
Results: Problems Fixed

Problems fixed

Error Models

Fraction of Problems fixed

F1 F2 F3 F4 F5

Array Reverse
Palindrome
Max
Factorial
isIncreasing
Sort
Results: Performance

Performance with Error Models

![Graph showing performance with error models. The x-axis represents error models (F1 to F5), and the y-axis represents running time (in s). The lines represent different error models: reverse, palindrome, max, isIncreasing, factorial, and sort. The graph shows trends in running times across different error models.]
Results: Generalization

Generalization of Error models

- Specific Error model
- Array reverse error model

<table>
<thead>
<tr>
<th>Error Model</th>
<th>Number of Fixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>palindrome</td>
<td>60</td>
</tr>
<tr>
<td>array max</td>
<td>70</td>
</tr>
<tr>
<td>isIncreasing</td>
<td>30</td>
</tr>
<tr>
<td>sort</td>
<td>10</td>
</tr>
</tbody>
</table>
Feedback for Tutoring System

Analysis

######## Running ProSE v1.02 ########

The solution requires 2 changes

Suggested Changes:

1. Change the conditional operator to < inside the loop conditional front > back.
2. Change the loop increment operator ++back to --.
Broad research agenda ahead

• Transformative for students in under-funded schools
  – Reduce the resources required to support quality instruction
  – Enable “true” distance education for programming courses

• Same technology could be used for automatic tutoring
  – Identify errors stemming from deep misconceptions
    (e.g. not understanding difference in values vs. references)
  – Synthesize small examples that make misconceptions explicit
Automatically Generating Problems for Online Courses in Embedded Systems

Dorsa Sadigh, UC Berkeley
Advisor: Prof. Sanjit Seshia
ExCAPE Kick-off meeting: Education & Outreach
June 5, 2012
Motivation

> Recent trend towards massive open online courses (MOOCs)

> Moving online from traditional classroom requires large scale:
  • Problem Generation
  • Grading & Feedback
  • Monitoring

> Given existing problems, we want to generate “Similar” problems
  • Similar concepts tested
  • Abstract problems to a template, generate similar instances
Intro. to Embedded Systems: EECS149

- This course is intended to introduce students to the design and analysis of computational systems that interact with physical processes.

http://leeseshia.org/
Example: Real-Time Scheduling

> This problem studies fixed-priority scheduling. Consider two tasks to be executed periodically on a single processor, where task1 has period $p_1=4$ and task2 has period $p_2=6$.

> Let the execution time of task1 be $e_1=1$. Find the maximum value for the execution time $e_2$ of task2 such that the RM schedule is feasible.
Consider \(<n>\) periodic/sporadic tasks, with execution times \(<e_{1,…,n}>\), periods \(<p_{1,…,n}>\), and deadlines \(<d_{1,…,n}>\) with fixed/dynamic priorities, with/without some precedence graph, on \(<m>\) processors. Compare different scheduling procedures \(<S>\) [RM,EDF,...], and their properties \(<\phi>\) [feasibility, processor utilization,...]
Formulate the Solution

> RM Schedule:

Two Tasks:

$$\left[ p_1 < p_2 \right] \rightarrow \left[ \frac{p_2}{p_1} \right] * e_1 + e_2 = p_2$$

Three Tasks:

$$\left[ p_1 < p_2 < p_3 \right] \rightarrow \left[ \frac{p_3}{p_1} \right] * e_1 + \left[ \frac{p_3}{p_2} \right] * e_2 + e_3 = p_3$$

N Tasks:

$$\left[ p_1 < \ldots < p_n \right] \rightarrow \left[ \frac{p_n}{p_1} \right] * e_1 + \ldots + \left[ \frac{p_n}{p_{n-1}} \right] * e_{n-1} + e_n = p_n$$
RM Scheduling as SMT

:formula ( 
  let (?m1 (bvsub p2 (bvmul p1 (bvudiv p2 p1))))
  (let (?m2 (bvsub p1 (bvmul p2 (bvudiv p1 p2)))))
  (let (?c1 (ite (= ?m1 bv0[8]))
    (bvudiv p2 p1)
    (bvadd bv1[8] (bvudiv p2 p1))))
  (let (?c2 (ite (= ?m2 bv0[8]))
    (bvudiv p1 p2)
    (bvadd bv1[8] (bvudiv p1 p2))))
  (let (?f1 (bvmul ?c1 e1))
  (let (?f2 (bvmul ?c2 e2))
  (let (?r1 (bvadd ?f1 e2))
  (let (?r2 (bvadd ?f2 e1))
  (and (implies (bvule p1 p2) (= ?r1 p2))
  (implies (not (bvule p1 p2)) (= ?r2 p1))))))))))

:logic QF_BV
p1 = bv4[8]
p2 = bv6[8]
e1 = bv1[8]

Find e2?
RM Scheduling Solution

**RM Schedule:**
SMT Solver finds $e_2$ is 4.
Now we can generate random $e_1$, $p_1$ and $p_2$, in order to find $e_2$. 

---

**Diagram:**
- Task 1: Bar from 0 to 2
- Task 2: Bar from 2 to 12

---

49
Another group of Problems

Property $\langle \phi \rangle$
- English/LTL

Synch/Asynch
- Formal Descrip./Code/Timed

Trace $\langle \psi \rangle$
- Witness/Counterexample

$\langle S \rangle$
- Time/Event Triggered

$\langle E \rangle$
- Time/Event Triggered
<table>
<thead>
<tr>
<th>Given</th>
<th>Find</th>
<th>Variation</th>
<th>Exercise #</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle \phi \rangle$</td>
<td>$\langle M \rangle$</td>
<td>Property in English LTL spec. M could be a timed automaton. Modify some pre-existing M.</td>
<td>3.1, 3.2, 3.3, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.8, 9.4, 9.6, 13.2, 13.3</td>
</tr>
<tr>
<td>$\langle M \rangle$</td>
<td>$\langle \psi \rangle$</td>
<td>Reachable trace, Describe output</td>
<td>3.3, 3.4, 3.5, 4.2</td>
</tr>
<tr>
<td>$\langle M \rangle$</td>
<td>$\langle \phi \rangle$</td>
<td>Model can be given in code or formal description</td>
<td>3.2, 12.3</td>
</tr>
<tr>
<td>$\langle M \rangle &amp; \langle \psi \rangle$</td>
<td>$\langle \phi \rangle$</td>
<td>Given input trace find output trace</td>
<td>9.5</td>
</tr>
<tr>
<td>$\langle M \rangle &amp; \langle \psi \rangle$</td>
<td>$\langle \phi \rangle$</td>
<td>Counterexample trace or reachable trace</td>
<td>4.3, 12.1</td>
</tr>
</tbody>
</table>
# How to Find the Solutions?

<table>
<thead>
<tr>
<th>Given</th>
<th>Find</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\phi&gt;)</td>
<td>(&lt;M&gt;)</td>
<td>Constrained Synthesis or Repair</td>
</tr>
<tr>
<td>(&lt;M&gt;)</td>
<td>(&lt;\psi&gt;)</td>
<td>Simulation of Model</td>
</tr>
<tr>
<td>(&lt;M&gt;)</td>
<td>(&lt;\phi&gt;)</td>
<td>Specification Mining</td>
</tr>
<tr>
<td>(&lt;M&gt; &amp; &lt;\psi&gt;)</td>
<td>(&lt;\psi&gt; \mid &lt;\phi&gt;)</td>
<td>Simulation with Guidance</td>
</tr>
<tr>
<td>(&lt;M&gt; &amp; &lt;\phi&gt;)</td>
<td>(&lt;\psi&gt;)</td>
<td>Model Checking</td>
</tr>
</tbody>
</table>
Future Work

• Implement the ideas for model based problems. So far we have looked at the examples that study a “model” of a state machine. Next step is to connect these to code.

• Bring the factor of creativity in automatic problem generation.

• Getting more data from the students this Fall
Education and Outreach

• 2013 Summer School
• Programs for High School Students
• Intelligent Tutoring Software
• Course Modules

• Others (from proposal): Annual workshop; integrative training, undergraduate education, diversity enhancement
Course Modules

- Target audience
- Covering the breadth of ExCAPE

- *From Summer School?*
- *Master plan?*
Education and Outreach

• Looking forward to involvement of all PIs
• Suggestions welcome
• Please let Steve and Stéphane know of your relevant activities