Intelligent Tutoring Systems
A Logical Perspective

Sumit Gulwani
Microsoft Research, Redmond

July 2012
Background: Program Synthesis

• Application domain
  - Software Developers
  - End-Users
  - Education

• Specification of intent
  - Logic
  - Examples
  - Natural Language

• Search techniques
  - SAT/SMT solvers (Formal Methods)
  - A*-style goal-directed search (AI)
  - Version space algebras (Machine Learning)

Value Proposition

• Interactive Feedback
  - Answer-scripts graded immediately.
  - Students provided feedback during solution generation

• Exploration
  - Interactive Visualization Tools
  - Simulation Tools

• Social
  - Students working on same problem at same time can come together.
  - Learning experience for past students can be used to benefit the learning experience for future students.
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team
Given a triangle $XYZ$, construct circle $C$ such that $C$ passes through $X$, $Y$, and $Z$. 

PLDI 2011: Gulwani, Korthikanti, Tiwari.
Formal specification of the problem

Given a triangle $XYZ$, construct circle $C$ such that $C$ passes through $X$, $Y$, and $Z$.

Precondition:
$\text{Slope}(X,Y) \neq \text{Slope}(X,Z) \land \text{Slope}(X,Y) \neq \text{Slope}(Z,X)$

Postcondition:
$\text{LiesOn}(X,C) \land \text{LiesOn}(Y,C) \land \text{LiesOn}(Z,C)$

Where $\text{LiesOn}(X,C) \equiv \text{Distance}(X,\text{Center}(C)) = \text{Radius}(C)$
Geometry Program: A straight-line composition of geometry methods.

Geometry Types: Point, Line, Circle

Geometry Methods:
- Ruler(Point, Point) → Line
- Compass(Point, Point) → Circle
- Intersect(Circle, Circle) → Pair of Points
- Intersect(Line, Circle) → Pair of Points
- Intersect(Line, Line) → Point
Given a triangle $XYZ$, construct circle $C$ such that $C$ passes through $X$, $Y$, and $Z$.

1. $C_1 = \text{Compass}(X,Y)$;
2. $C_2 = \text{Compass}(Y,X)$;
3. $<P_1,P_2> = \text{Intersect}(C_1,C_2)$;
4. $L_1 = \text{Ruler}(P_1,P_2)$;
5. $D_1 = \text{Compass}(Z,X)$;
6. $D_2 = \text{Compass}(X,Z)$;
7. $<R_1,R_2> = \text{Intersect}(D_1,D_2)$;
8. $L_2 = \text{Ruler}(R_1,R_2)$;
9. $N = \text{Intersect}(L_1,L_2)$;
10. $C = \text{Compass}(N,X)$;
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team
Let $L$ be the language containing all strings over \{a,b\} that have \textbf{the same number of occurrences of “ab” as occurrences of “ba”}. Construct an automata that accepts $L$, or prove that $L$ is non-regular.

"Regular", <Automata for $L$>
Let $L$ be the language containing all strings over $\{a, b\}$ that have the same number of occurrences of “$a$” as occurrences of “$b$”. Construct an automata that accepts $L$, or prove that $L$ is non-regular.

“Non-regular”, <Proof of non-regularity>
Intelligent Tutoring Systems

- Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

- Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    - Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    - Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    - Interest from Office Education Team

Joint work with Jain, Singh, Sharang, Sharma, Karkare, Roy
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team
Solution Generation by Example

Division method

Example 2: Find the LCM of 144, 96, 160

Step 1: Write the numbers as shown. Divide all numbers by a prime number which divides at least two of the numbers.

Step 2: Write the quotient in each case below the number.
If a number cannot be divided exactly, write the number as it is in the next row.

Step 3: Keep dividing by prime numbers until the last row has coprime numbers with no common factors.

Step 4: Multiply all divisors and all numbers left in the last row. The product gives the LCM of the given numbers.

LCM = 2 × 2 × 2 × 2 × 2 × 3 × 3 × 5 = 1440,

Exercise 3.4

A. Find the LCM of the following sets of numbers.

1) 108, 144
4) 98, 147
7) 76, 57
10) 75, 90, 125
13) 48, 60, 84
16) 255, 340, 765, 425

2) 72, 90
5) 105, 70
8) 252, 144
11) 93, 62, 120
14) 65, 115, 130

3) 39, 195
6) 85, 51
9) 256, 64
12) 75, 90
15) 39, 52, 65
Intelligent Tutoring Systems

- **Various Aspects**
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

- **Various Domains**
  - Geometry, Algebra, Physics, Chemistry, Logic
    - Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    - Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    - Interest from Office Education Team
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team
Background: PexForFun

This puzzle is an interactive Coding Duel. Can you write code that matches a secret implementation? Other people have already won this Duel 618 times! Help

using System;

public class Program
{
    public static bool Puzzle(string s)
    {
        // Can you write code that determines if the string is an anagram?
        return false;
    }
}
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;
    }
}
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;
    }
}
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;
    }
}
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

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;
}
}

Same as initial attempt except Console.WriteLine!
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;
    }
}

Buggy Program for Array Reverse

No change! Sign of Frustation?
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

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!
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;
    }
}

**Buggy Program for Array Reverse**

```
Almost correct! (a[i-1] instead of a[i] in loop body)
```
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;
    }
}
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;
    }
}

Back to bigger error!
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! Frustation!
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

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.
Proposal: Semantic Grading

Provides additional value over counterexample feedback.

• More friendly feedback.
  - Helpful for students who give up after several tries (with only counterexample feedback).

• Grading
  - Counterexample feedback does not distinguish between a slightly incorrect solution and one that is very far off from being correct.
Example Errors

Array Index: \(v[a] \rightarrow v[[a+1, a-1]]\)

Increment: \(v++ \rightarrow \{ ++v, v--, --v \}\)

Conditional: \(a \text{ op } b \rightarrow a \text{ ops } b\)

where \(\text{ops} = \{ <, >, <=, >=, ==, != \}\)

Initialization: \(v=n \rightarrow v=[n+1, n-1, 0]\)

Return Value: \(\text{return } v \rightarrow \text{return } ?v\)

Joint work with Rishabh Singh and Armando Solar-Lezama
Automated Grading (Array Reverse)

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;
    }
}

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];
            ++back;
            ++front;
            temp = a[back];
        }
        return a;
    }
}
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team
Example Problem: \((\sec x + \cos x) (\sec x - \cos x) = \tan^2 x + \sin^2 x\)

Query: \((T_1(x) \pm T_2(x)) (T_3(x) \pm T_4(x)) = T_5^2(x) \pm T_6^2(x)\)
\[T_1 \neq T_5\]

New problems generated:
\((\csc x + \cos x)(\csc x - \cos x) = \cot^2 x + \sin^2 x\)
\((\csc x - \sin x)(\csc x + \sin x) = \cot^2 x + \cos^2 x\)
\((\sec x + \sin x)(\sec x - \sin x) = \tan^2 x + \cos^2 x\)
Algebra Problem Generation

Example Problem

Query Generation

Query

Query Execution

Query Refinement

Refined Query

Results OK?

New Problems

Yes

No

Similar Problems
Limits/Series Problem

Example Problem: \[
\lim_{n \to \infty} \sum_{i=0}^{n} \frac{2i^2 + i + 1}{5^i} = \frac{5}{2}
\]

Query: \[
\lim_{n \to \infty} \sum_{i=0}^{n} \frac{C_0 i^2 + C_1 i + C_2}{C_3^i} = \frac{C_4}{C_5}
\]

\[C_0 \neq 0 \land \gcd(C_0, C_1, C_2) = \gcd(C_4, C_5) = 1\]

New problems generated:

\[
\lim_{n \to \infty} \sum_{i=0}^{n} \frac{3i^2 + 2i + 1}{7^i} = \frac{7}{3}
\]

\[
\lim_{n \to \infty} \sum_{i=0}^{n} \frac{3i^2 + 3i + 1}{4^i} = 4
\]

\[
\lim_{n \to \infty} \sum_{i=0}^{n} \frac{i^2}{3^i} = \frac{3}{2}
\]

\[
\lim_{n \to \infty} \sum_{i=0}^{n} \frac{5i^2 + 3i + 3}{6^i} = 6
\]
Example Problem: \[ \int (\csc x) (\csc x - \cot x) \, dx = \csc x - \cot x \]

Query: \[ \int T_0(x)(T_1(x) \pm T_2(x))dx = T_4(x) \pm T_5(x) \]

\[ T_1 \neq T_2 \land T_4 \neq T_5 \]

New problems generated:

\[ \int (\tan x) (\cos x + \sec x) \, dx = \sec x - \cos x \]

\[ \int (\sec x) (\tan x + \sec x) \, dx = \sec x + \cot x \]

\[ \int (\cot x) (\sin x + \csc x) \, dx = \sin x - \csc x \]
Determinant Problem

Ex. Problem

\[
\begin{pmatrix}
(x + y)^2 & zx & zy \\
zx & (y + z)^2 & xy \\
yz & xy & (z + x)^2 \\
\end{pmatrix} = 2xyz (x + y + z)^3
\]

Query

\[
\begin{pmatrix}
F_0(x, y, z) & F_1(x, y, z) & F_2(x, y, z) \\
F_3(x, y, z) & F_4(x, y, z) & F_5(x, y, z) \\
F_6(x, y, z) & F_7(x, y, z) & F_8(x, y, z) \\
\end{pmatrix} = C_{10} F_9(x, y, z)
\]

\[F_i := F_j[x \to y; y \to z; z \to x] \text{ where } (i, j) \in \{(4,0), (8,4), (5,1), \ldots \}\]

New problems generated:

\[
\begin{pmatrix}
y^2 & x^2 & (y + x)^2 \\
(z + y)^2 & z^2 & y^2 \\
z^2 & (x + z)^2 & x^2 \\
\end{pmatrix} = 2(xy + yz + zx)^3
\]

\[
\begin{pmatrix}
yz + y^2 & xy & xy \\
yz & zx + z^2 & yz \\
zx & zx & xy + x^2 \\
\end{pmatrix} = 4x^2 y^2 z^2
\]
Intelligent Tutoring Systems

• **Various Aspects**
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• **Various Domains**
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - **Language Learning**
    • Interest from Office Education Team
Sentence Completion Problems

1. The principal characterized his pupils as __________ because they were pampered and spoiled by their indulgent parents.

2. The commentator characterized the electorate as __________ because it was unpredictable and given to constantly shifting moods.

(a) cosseted
(b) disingenuous
(c) corrosive
(d) laconic
(e) mercurial
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team
\[
\tan 3x \tan 2x \tan x = \tan 3x - \tan 2x - \tan x
\]

\[
\begin{pmatrix}
yz - x^2 & zx - y^2 & xy - z^2 \\
zx - y^2 & xy - z^2 & yz - x^2 \\
xy - z^2 & yz - x^2 & zx - y^2
\end{pmatrix}
\]

\[
\begin{pmatrix}
A_1 \sin^3 \alpha & B_1 \sin^3 \beta & C_1 \sin^3 \gamma \\
A_2 \sin \alpha & B_2 \sin \beta & C_2 \sin \gamma
\end{pmatrix}
\]

Joint work with Alex Polozov and Sriram Rajamani
Prove \((\csc x - \sin x)(\sec x - \cos x)(\tan x + \cot x) = 1\)

\[
\text{L.H.S.} = \left(\frac{1}{\sin x} - \sin x\right)\left(\frac{1}{\cos x} - \cos x\right)\left(\frac{\sin x}{\cos x} + \frac{\cos x}{\sin x}\right)
\]

\[
= \left(\frac{1 - \sin^2 x}{\sin x}\right)\left(\frac{1 - \cos^2 x}{\cos x}\right)\left(\frac{\sin^2 x + \cos^2 x}{\cos x \sin x}\right)
\]

\[
= \left(\frac{\cos^2 x}{\sin x}\right)\left(\frac{\sin^2 x}{\cos x}\right)\left(\frac{1}{\cos x \sin x}\right)
\]

\[
= 1
\]
Intelligent Tutoring Systems

• Various Aspects
  - Solution Generation
  - Grading/Feedback
  - Problem Generation
  - Content Entry

• Various Domains
  - Geometry, Algebra, Physics, Chemistry, Logic
    • Partnership with UW and IITK for deployment in schools
  - Programming, Automata, Excel, SQL
    • Partnership with MIT, UPenn, UIUC for EdX/internal deployment
  - Language Learning
    • Interest from Office Education Team

CHI 2012: Cheema, Gulwani, LaViola
Conclusion

• **Technical Perspective**
  - **Aspects**: Solution+Problem Generation, Grading, Content Entry
  - **Domains**: Math/Science, Programming, Language Learning

• **Value Proposition:**
  - **Short term**: Interactive Feedback, Exploration, Social
  - **Long Term**: Ultra-intelligent computer, Model of human mind, Inter-stellar travel 😊

• **Crowdsourcing Opportunities**
  - **Solution Generation**: Natural Language to Logic translation
  - **Problem Generation**: Rate difficulty and ambiguity level
  - **Automated Grading**: Provide natural explanations for errors