Automated Grading of DFA Constructions

Rajeev Alur (Penn), Loris D’Antoni (Penn), Sumit Gulwani (MSR), Bjoern Hartmann (Berkeley), Dileep Kini (UIUC), Mahesh Viswanathan (UIUC)
Motivations

• Grading is a tedious and time consuming task
• Human grades are often inconsistent
• MOOCs (Massive Online Open Courses) admits thousands of students, infeasible to grade manually
Draw the DFA accepting the language:

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

Solution:
Student Solutions
The problem description was

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

The student instead drew DFA for

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

**INTUITION**: find the distance between the two language descriptions
Mosel: MSO + Syntactic Sugar

Mosel: similar to MSO; predicates describing DFAs

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

• If we had such descriptions we could use tree edit distance to check how far they are from each other
• “Easy” to go from such Mosel predicates to automata (classical MSO to DFA algorithm)
• However, what we need is:

  \textit{given a DFA compute a (small) Mosel formula}
Brute Force Search

Enumerate all the predicates and check for equivalence with target DFA

Search pruning and speeding:

• Avoid trivially equivalent predicates (A V B, B V A)
• Approximate equivalence using set of test strings:
  – Generate sets of positive and negative examples that distinguish each state in the target DFA
  – One can prove all such strings are enough to prove inequivalent all DFAs of smaller size than target DFA
Solution Syntactic Mistake

The student forgot one final state

INTUITION: find the smallest number of syntactic modification to fix solutions
DFA Edit Difference

Compute DFA edit distance:
  – Number of edits necessary to transform the DFA into a correct one

An edit is
  – Make a state (non)final
  – Add a new state
  – Redirect a transition
DFA Edit Difference: How to compute it?

We try every possible edit and check for equivalence

• Speed up equivalence by using test set
• The problem of finding DFAED is in NP (is it NP-hard?)
Solution Semantic Mistake

The student didn’t see that the ‘a’ loop might not be traversed

INTUITION: find on how many strings the student is wrong
Approximate Density

S = correct solution  A = student attempt

Compute Symmetric Difference: \( D = S \setminus A \cup A \setminus S \)

- Measure relative size of \( D \) with respect to \( S \)
  \[ \text{Size}(D,S) = \lim_{n \to \infty} \frac{D^n}{S^n} \]

- Size\((D,S)\) is not computable in general (the limit oscillates)

- Approximate the limit to finite \( n \)
Evaluation 1/2

H1, H2 = human graders
N = naïve grader
T = tool

Tool is closer to humans than humans are to each other
Evaluation 2/2

H1, H2 = human graders
N = naïve grader
T = tool

Tool and humans look indistinguishable

<table>
<thead>
<tr>
<th>Problem</th>
<th>Attempts</th>
<th>Average</th>
<th>Standard Deviation</th>
<th>Pearson Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tot. Dis.</td>
<td>H1-H2</td>
<td>H1-T</td>
<td>H1-N</td>
</tr>
<tr>
<td>(L_1 = {s \mid s \text{ starts with } a \text{ and has odd number of } ab \text{ substrings}})</td>
<td>131 108</td>
<td>0.99</td>
<td>0.54</td>
<td>0.22</td>
</tr>
<tr>
<td>(L_2 = {s \mid s \text{ has more than } 2 \text{ } a's \text{ or more than } 2 \text{ } b's})</td>
<td>110 100</td>
<td>-0.66</td>
<td>0.85</td>
<td>0.26</td>
</tr>
<tr>
<td>(L_3 = {s \mid s \text{ where all odd positions contain the symbol } a})</td>
<td>96 75</td>
<td>-0.52</td>
<td>0.86</td>
<td>0.26</td>
</tr>
<tr>
<td>(L_4 = {s \mid s \text{ begins with } ab \text{ and }</td>
<td>s</td>
<td>\text{ is not divisible by } 3})</td>
<td>92 68</td>
<td>0.40</td>
</tr>
<tr>
<td>(L_5 = {s \mid s \text{ contains the substring } ab \text{ exactly twice}})</td>
<td>52 46</td>
<td>0.02</td>
<td>0.19</td>
<td>0.29</td>
</tr>
<tr>
<td>(L_6 = {s \mid s \text{ contains the substring } aa \text{ and ends with } ab})</td>
<td>38 31</td>
<td>-0.50</td>
<td>-1.34</td>
<td>-1.5</td>
</tr>
</tbody>
</table>
Pro’s and Cons

Pros:
• On disagreeing cases, human grader often realized that his grade was inaccurate
• Identical solutions receive same grades and correct attempts awarded max score (unlike human)

Cons:
• For now limited to small DFAs
• When two types of mistakes happen at same time, the tool can’t figure it out
Conclusions

AutomataTutor: a tool that grades DFA constructions fully automatically

Few new automata problems:

• How to compute DFA edit difference?
• How to synthesize Mosel formulas in a better way?
• How to compute language sizes in a way that is always defined and accurate?
Questions?

Thank you!