Milestone 3 Progress Evaluation

Automatic Program Feedback

1.

Team Members:

Andre Leone  Ryan Hartman  Calvin Winget
aleone2013@my.fit.edu  rhartman2014@my.fit.edu  cwinget2014@my.fit.edu

Faculty sponsors:

Dr. Philip Chan  Dr. Ryan Stansifer
pke@cs.fit.edu  ryan@cs.fit.edu
2. Progress of current milestone (progress matrix)

<table>
<thead>
<tr>
<th>Task</th>
<th>Completion %</th>
<th>Andre Leone</th>
<th>Ryan Hartman</th>
<th>Calvin Winget</th>
<th>To do</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Build version 2 of the feedback program</td>
<td>100%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>none</td>
</tr>
<tr>
<td>2. Evaluate Results</td>
<td>100%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>none</td>
</tr>
<tr>
<td>3. Analyze incorrect feedback</td>
<td>100%</td>
<td>30%</td>
<td>35%</td>
<td>35%</td>
<td>none</td>
</tr>
<tr>
<td>4. Brainstorm possible updates that can be done to improve the feedback results</td>
<td>100%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>none</td>
</tr>
<tr>
<td>5. Implement updates</td>
<td>100%</td>
<td>40%</td>
<td>30%</td>
<td>30%</td>
<td>none</td>
</tr>
</tbody>
</table>

3. Discussion of Tasks

- Task 1: We added in the ability to instrument calls to “add” and “remove”, in addition to “get” and “set”. This allows us to conduct experiments on a wider range of data. It does not limit us to certain types of List problems. Also, we made minor enhancements to the existing program, such as two spelling errors and added .config file features.
- Task 2:
  - Experiment 1 (Selection Sort):
    - 14 reference programs and 28 submitted programs
      - Each reference program with 1 type of bug has two similar submitted programs of the same type of bug
    - 20/28 (71.4%) of submitted programs were correctly matched to a reference program with the same type of bug (feedback)
    - With 95% confidence we can say that the percentage of correctly matched submitted programs will be between 54.2% and 87.8%. Given
that our lower bound is greater than 50%, we can say that these results are statistically significant.

- Bias does exist in these results because we manually created the bug variants. Are programs may have been unrealistically similar. In reality, we do not actually know how much similar submissions with the same bug would differ.
  - Experiment 2 (Hackerrank Java List):
    - 56/80 (70%) of submitted programs were matched to a reference program with the same Hackerrank score as the submitted program.
    - Given the score distribution of this experiment, the expected number to randomly match correctly is 34 submitted programs. The probability that 56 or greater submitted programs were matched by random chance is 0.000000009%.
  - Experiment 3 (Hackerrank Java ArrayList)
    - 29/57 (50.88%) of submitted programs were matched to a reference program with the same Hackerrank score as the submitted program.
    - Given the score distribution of this experiment, the expected number to randomly match correctly is 13 submitted programs. The probability that 29 or greater submitted programs were matched by random chance is 0.0002%.
  - We concluded that trace files are useful. We are able to explain Hackerrank scores using trace files.

- Task 3: After plotting the results for both experiment 2 and experiment 3 on graphs analyzing the edit distance metric, we found that when the edit distance is greater than 0, there is NOT a strong correlation between edit distance and behavioral similarity between two programs. Therefor, trace files contain useful information but the edit distance metric itself could be improved upon.

- Task 4:
  - A. Line numbers in the trace files could be inflating the edit distances in an unexpected manner
  - B. The array state itself may be the only thing needed as it changes throughout the run time of the program
  - C. Perhaps we don't need the line numbers or array state at all. Instead we only need the method name that was called and the parameter that was passed to it.

- Task 5:
  - We have implemented the possible ideas that we brainstormed to test which method of instrumentation leads to the best results. Experiments 4, 5, and 6 were all done on the Hackerrank Java ArrayList question.
  - Experiment 4 (Task 4 - A)
    - 33/57 (57.89%) of submitted programs were matched to a reference program with the same Hackerrank score as the submitted program.
    - Given the score distribution of this experiment, the expected number to randomly match correctly is 13 submitted programs. The probability that
33 or greater submitted programs were matched by random chance is 0.0000005%.

- **Experiment 5 (Task 4 - B)**
  - 8/57 (14.04%) of submitted programs were matched to a reference program with the same Hackerrank score as the submitted program.
  - Given the score distribution of this experiment, the expected number to randomly match correctly is 13 submitted programs. The probability that 8 or greater submitted programs were matched by random chance is 97.08%.

- **Experiment 6 (Task 4 - C)**
  - 26/57 (45.61%) of submitted programs were matched to a reference program with the same Hackerrank score as the submitted program.
  - Given the score distribution of this experiment, the expected number to randomly match correctly is 13 submitted programs. The probability that 26 or greater submitted programs were matched by random chance is 0.0088%.

### 4. Discussion of contributions

- **Andre Leone:**
  - Contributed to the version 2 updates
  - Created the probability program that computes the probability randomly getting x or greater programs correctly matched to Hackerrank programs of the same point value.
  - Plotted the edit distance graphs to see if edit distance is working in the way that we would expect. Found that it is not working as we would expect and in reality most of our correct matches come from things that have an edit distance of 0. Concluded that our current edit distance is unreliable greater than 0 at this time.
  - Ran the experiments that we set up.

- **Ryan Hartman:**
  - Analyzed Selection sort results. Went through the incorrectly matched submitted programs to figure out why those programs didn’t match to their expected reference program.
  - Analyzed the Hackerrank experiment results from experiments 2 and 3 to determine possible causes of the programs that matched to reference programs of incorrect point values.
  - Determined possible experiments that could be run in Task 4.

- **Calvin Winget:**
  - Created extra selection sort submission programs so that there were at least 2 submissions that matched each reference program.
  - Fixed many of the programs getting their input from command line arguments.
- Analyzed Selection sort results. Went through the incorrectly matched submitted programs to figure out why those programs didn’t match to their expected reference program.
- Discussed methods to evaluate statistical significance with Dr. Bond

### 5. Plan for the next Milestone (task matrix)

<table>
<thead>
<tr>
<th>Task</th>
<th>Andre Leone</th>
<th>Ryan Hartman</th>
<th>Calvin Winget</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Build version 3 of the feedback program</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
<tr>
<td>2. Evaluate Results</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
<tr>
<td>3. Analyze incorrect feedback</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
<tr>
<td>4. Brainstorm possible updates that can be done to improve the feedback results</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
</tbody>
</table>

### 6. Discussion of each planned task for the next milestone

- **Task 1:**
  - Figure out what is actually included in the edit distance calculations
  - Incorporate more information in the edit distance calculations from the trace files. Our current ideas are adding the array state and index.
  - Add checks outside of edit distance. For example a check for no output, an infinite loop (timeout) check, and some way to see if the program is just spitting out nonsense to try and fool the edit distance metric. Possibly lines of code or another program size metric.
- **Task 2:** Make results easier to evaluate by printing out a matrix of the edit distance between our programs.
- **Task 3:** Pay extra attention to programs where the submissions incorrectly match to the reference programs. Try to determine what is wrong with our methodology that would allow this to happen.
- **Task 4:** Brainstorm ways to avoid the incorrect matches our system is making.
7. Sponsor feedback on each task for the current milestone

- Task 1:

- Task 2:

- Task 3:

- Task 4:

Sponsor Signature: ____________________________ Date: __________________
8. Sponsor Evaluation
- Sponsor: Detach and return this page to Dr. Shoaff
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or right down a real number between 0 and 10)

<table>
<thead>
<tr>
<th>Name</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>5.5</th>
<th>6</th>
<th>6.5</th>
<th>7</th>
<th>7.5</th>
<th>8</th>
<th>8.5</th>
<th>9</th>
<th>9.5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andre Leone</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5.5</td>
<td>6</td>
<td>6.5</td>
<td>7</td>
<td>7.5</td>
<td>8</td>
<td>8.5</td>
<td>9</td>
<td>9.5</td>
<td>10</td>
</tr>
<tr>
<td>Calvin Winget</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5.5</td>
<td>6</td>
<td>6.5</td>
<td>7</td>
<td>7.5</td>
<td>8</td>
<td>8.5</td>
<td>9</td>
<td>9.5</td>
<td>10</td>
</tr>
<tr>
<td>Ryan Hartman</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5.5</td>
<td>6</td>
<td>6.5</td>
<td>7</td>
<td>7.5</td>
<td>8</td>
<td>8.5</td>
<td>9</td>
<td>9.5</td>
<td>10</td>
</tr>
</tbody>
</table>

- Sponsor Signature: ________________________________  Date: ______________