A Criterion for Filtering Code Clone Related Bugs

Yasuhiro Hayase†, Y. Yong Lee, Katsuro Inoue†
Osaka University

†{y-hayase,inoue}@ist.osaka-u.ac.jp
Background

- **Code Clone**
  - a code fragment occurring more than once in identical or similar form into a software system
  - introduced in the source program because of various reasons such as reusing code by `copy-and-paste`

- **Clone Pair**
  - a pair of code fragment that are identical or similar each other
Clone-Related Bugs

- Clone-Related Bug
  - Clones are often modified after copy-and-paste
  - Faults are possibly introduced through the modification.

- CP-Miner (Zhenmen Li et al., 2007)
  - Detecting clone-related bugs
    - Find and present inconsistent renaming of identifier between clones
  - Problem
    - False-positives
      - Propose a new criterion to filter out the false-positives
Brief Summary of CP-Miner

- Map identifier appearances between a clone pair
- Create renaming table between a clone pair
- Compute **UnchangedRatio (UR)**
  - Smaller UR except for 0 means that the renaming is suspicious.

<table>
<thead>
<tr>
<th>ID in C1</th>
<th>ID in C2</th>
<th>count</th>
<th>UR</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>p</td>
<td>c</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>p</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>y</td>
<td>q</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>z</td>
<td>x</td>
<td>2</td>
<td>0.20</td>
</tr>
</tbody>
</table>

```
... = b;
b++;  
for (p=0; p<10; p++) {
  x += p;
}

a(x);
a(x);

... = c;
c++;  
for (q=0; q<10; q++) {
  y += p;
}

a(x);
a(z);
```

Leave unchanged
Our Approach: New Filtering Criterion

- x seems intentionally renamed to different symbols
- Criterion **Conflict (CF)**
  - true if the identifier mapped into two or more other identifiers
  - false otherwise

<table>
<thead>
<tr>
<th>ID in C1</th>
<th>ID in C2</th>
<th>count</th>
<th>UR</th>
<th>CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>2</td>
<td>1</td>
<td>false</td>
</tr>
<tr>
<td>b</td>
<td>c</td>
<td>2</td>
<td>0</td>
<td>false</td>
</tr>
<tr>
<td>p</td>
<td>p</td>
<td>1</td>
<td></td>
<td>false</td>
</tr>
<tr>
<td>q</td>
<td></td>
<td>3</td>
<td></td>
<td>false</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
<td>2</td>
<td>0.20</td>
<td>true</td>
</tr>
<tr>
<td>x</td>
<td>z</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Implementation

- The filter using **Conflict** is implemented into clone-related bug detection system.

  The system uses CCFinder.
Evaluation

- An experiment was performed for evaluation
  - Target: arch module of Linux 2.6.6
  - Inconsistencies whose UnchangedRatio $\leq 0.4$ are reviewed in ascent order of UnchangedRatio

- Result
  - Reduce 27% of inconsistencies
  - Filter removes NO true-faults
Conclusion

- We proposed a criterion for filtering false-candidates detected by CP-Miner
  - The criterion recognizes the identifiers renamed to two or more names different from the original name
- The filter using the criterion is implemented into a clone-related bug detection system
- The filter reduces 27% of inconsistencies