Motivation

- Programs hold a lot of rules
- Source code contains useful information for inferring these rules
- Data mining is a suitable technique to extract information from large software source code

Projects

- CP-Miner
  - Detecting copy-paste code and related bugs
  - Published in OSDI’04
- PR-Miner
  - Extracting programming rules and detecting violations
  - Published in ESEC/FSE’05
  - Both CP-Miner and PR-Miner are under being commercialized.

Overview of CP-Miner

- Copy-pasting is common in large software
- Copy-pasted code is error prone
- CP-Miner
  - Can efficiently identify copy-paste in large software
  - Can identify copy-pasted code with modifications
  - Can detect copy-paste related bugs

  ```
  for (i=0; i<n; i++) {
    total[i].adr = list[i].addr;
    total[i].bytes = list[i].size;
    total[i].more = &total[i+1];
  }
  for (i=0; i<n; i++) {
    taken[i].adr = list[i].addr;
    taken[i].bytes = list[i].size;
    taken[i].more = &taken[i+1];
  }
  ```

  Example: A copy-paste related bug detected by CP-Miner in Linux 2.6.6

How Does CP-Miner Work?

- Identifying Copy-Pasted Code
  - Characteristics of copy-pasted code
    - Code based
    - Appears for at least twice
    - May be not exactly the same
  - Frequent sequence mining

- Detecting Copy-Paste Related Bugs
  - Forget-to-change bugs
  - Method: measure unchanged ratio

Overview of PR-Miner

- Programs follow many programming rules
- Most rules are implicit and undocumented
- Many rules can be very complex
- Violations lead to errors

  ```
  struct scsi_id_instance_data *sbp2_alloc_device ( … )
  {
    scsi_host = host_alloc ( … );
    if (!add_host (scsi_host, &pdev->dev))
      // scan_host (scsi_host) is missing!
  }
  ```

  Example: Violations detected in Linux 2.6.11.

How Does PR-Miner Work?

- Extracting programming rules
  - Idea: finding association among elements that are frequently used together in source code
  - Frequent itemset mining

  ```
  int twa_probe(struct pci_dev *pdev, ...
  {
    Scsi_Host 92
    host_alloc 92
    host_alloc 39
    host = host_alloc ( … );
    add_host (host, &pdev->dev);
    Scsi_Host 68
    pdev.dev 56
    Scsi_Host 36
    Scsi_Host 92
  }
  ```

  Example: Violations detected in Linux 2.6.11.

- Detecting violations
  - For violations of a programming rule
    - The rule holds for most cases
    - The rule is violated for a few cases
    - Confident > threshold
    - The rule is violated for a few cases
      - Confidence > 100%

Results of CP-Miner

- Bugs detected by CP-Miner were reported by us and are now fixed afterwards

<table>
<thead>
<tr>
<th>Software</th>
<th>#LOC</th>
<th>Copy-paste</th>
<th>Time (sec)</th>
<th>#Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>4.4 M</td>
<td>22.3%</td>
<td>20 min</td>
<td>49</td>
</tr>
<tr>
<td>FreeBSD</td>
<td>3.3 M</td>
<td>20.4%</td>
<td>20 min</td>
<td>31</td>
</tr>
<tr>
<td>Apache</td>
<td>224 K</td>
<td>17.7%</td>
<td>15 sec</td>
<td>5</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>458 K</td>
<td>22.2%</td>
<td>38 sec</td>
<td>2</td>
</tr>
</tbody>
</table>

- Modifications in copy-pasted segments
  - 65% have identifier renaming
  - 25% contain statement insertion/modification

Results of PR-Miner

- Rules extracted and bugs detected
  - 88-97% rules involve variables
  - Manually inspected the top 60 violations

<table>
<thead>
<tr>
<th>Software</th>
<th>#Rules</th>
<th>Time (sec)</th>
<th>#Bugs Verified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>32,283</td>
<td>42</td>
<td>16</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>6,128</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Apache</td>
<td>283</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Rule size
  - Only 14% of rules contain 2 elements
  - Most rules contain more than 2 elements