Translating Code Comments to Procedure Specifications

A. Blasi, A. Goffi, K. Kuznetsov, A. Gorla, M.D. Ernst, M. Pezzè, S. D. Castellanos

arianna.blasi@usi.ch

Wanted: missing specification

Specifications help test case generation:
- ✓ act as assertions
- ✓ filter invalid tests

But formal specifications are:
- ❌ seldom available
- ❌ hard to write

What we do have: Javadoc comments

```java
/**
 * @param vertex, must not be null
 * @return list of vertex's neighbors, not empty
 * @throws IllegalArgumentException if vertex is not found in the graph
 */
public List neighborsOf(V vertex){…}
```

**Pattern match**
- @param vertex, must not be null
  - vertex != null

**Syntax match**
- @return list of vertex neighbors, not empty
  - !result.isEmpty()

**Semantic match**
- @throws IllegalArgumentException if vertex is not found in the graph
  - !graph.contains(vertex) -> IllegalArgumentException

**Evaluation**

Jdoctor outperforms the state of the art

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>@tComment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toradocu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jdoctor</td>
<td>100%</td>
<td>80%</td>
</tr>
</tbody>
</table>

6 projects
829 translations

**Application**

Jdoctor improves Randoop's output by revealing more:

- False alarms
- Missed alarms
- New tests
- Invalid tests

Jdoctor improves Randoop's output by revealing more:

- False alarms
- Missed alarms
- New tests
- Invalid tests