

Estudo e desenvolvimento de novas funcionalidades do Java Pathfinder

MAC499 – Trabalho de Formatura Supervisionado

Orientadora: Ana Cristina Vieira de Melo (MAC-IME)

Rodrigo Della Vittoria Duarte

Bacharelado em Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

Introdução

- Alta demanda por software
- Orientação a Objetos largamente utilizado, especialmente Java
- Estrutura de Tratamento de Exceções vem ganhando espaço no desenvolvimento de novos software
- Poucas estratégias automáticas de testes e validação para sistemas orientados a objetos com tratamento de exceções

Problema

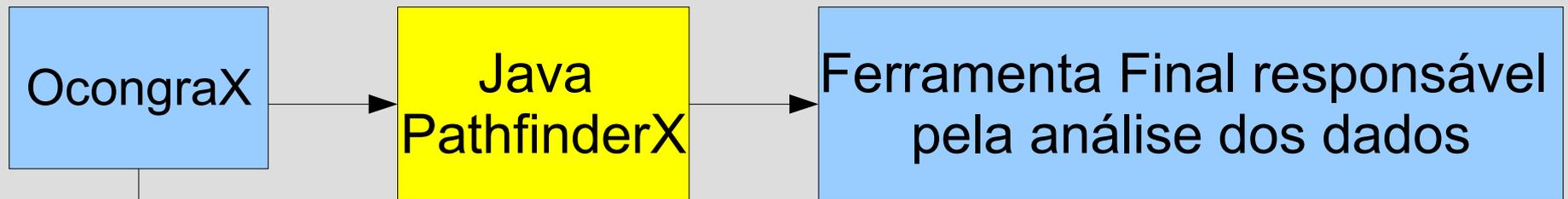
- A área de Testes e Validação consome muitos recursos
- Necessidade de escrever testes para todos pontos do software
- Podem surgir problemas oriundos de testes mal planejados ou executados
- Pequenas alterações no software exigem muitas alterações nos testes

Solução

- Utilizar Verificação Formal para diminuir casos de testes
- Pares de definição-uso de objetos extraídos durante a verificação
- Java Pathfinder: ferramenta já existente que verifica se um software satisfaz ou não certas propriedades
- Implementar nova funcionalidade que extraia informações acerca do comportamento do software durante a execução do Java Pathfinder
- Pontos cobertos pelo verificador não precisam ser novamente testados

Projeto Maior

- Este projeto é parte de um projeto maior da área de Engenharia de Software



(Geração do arquivo com os pares de definição e uso dos objetos)

Java PathfinderX

- Verificador de modelo desenvolvido pela Nasa para garantir certas propriedades do programa alvo
- Recebe um programa Java e um arquivo contendo todos os pares de definição-uso dos objetos do programa e devolve arquivos com todos pares cobertos e não cobertos dos objetos do programa alvo, além de uma listagem com todas instruções executadas
- Análise dos resultados permite diminuir o número de casos de testes necessários para validar o software

Exemplo Java Pathfinder

Programa Java

```
public class DeadlockException {
    static Lock lock1;
    static Lock lock2;
    static int state;

    public static void main(String[] args) {
        lock1 = new Lock();
        lock2 = new Lock();
        Process1 p1 = new Process1();
        Process2 p2 = new Process2();
        p1.start();
        p2.start();
    }

    public static void geraException() throws Exception {
        int valor = (int) (Math.random()*10);
        if(valor%2 == 0)
            throw new Exception();
    }
}

class Process1 extends Thread {
    public void run() {
        DeadlockException.state++;
        synchronized (DeadlockException.lock2) {
            synchronized (DeadlockException.lock1) {
                DeadlockException.state++;
            }
        }
    }
}

class Process2 extends Thread {
    public void run() {
        DeadlockException.state++;
        synchronized (DeadlockException.lock1) {
            synchronized (DeadlockException.lock2) {
                DeadlockException.state++;
            }
        }
    }
}
```

Java Pathfinder

Saída na Tela

```
----- transition #14 thread: 2
gov.nasa.ipf.jvm.choice.ThreadChoiceFromSet (Thread-0,>Thread-1)
  DeadlockException.java:58      : synchronized (DeadlockException.lock1) {
----- transition #15 thread: 2
gov.nasa.ipf.jvm.choice.ThreadChoiceFromSet (Thread-0,>Thread-1)
  DeadlockException.java:58      : synchronized (DeadlockException.lock1) {
  DeadlockException.java:59      : synchronized (DeadlockException.lock2) {
===== snapshot #1
thread index=1,name=Thread-
0,status=BLOCKED,this=Process1@223,priority=5,lockCount=0
  owned locks:Lock@219
  blocked on: Lock@218
  call stack:
    at Process1.run(DeadlockException.java:39)

thread index=2,name=Thread-
1,status=BLOCKED,this=Process2@248,priority=5,lockCount=0
  owned locks:Lock@218
  blocked on: Lock@219
  call stack:
    at Process2.run(DeadlockException.java:59)
===== results
error #1: gov.nasa.ipf.jvm.NotDeadlockedProperty "deadlock encountered: thread
inde..."
===== search finished: 06/11/07
17:18
```

Exemplo Java PathfinderX

Programa Java + Arquivo OcongraX

```
public class DeadlockException {
    static Lock lock1;
    static Lock lock2;
    static int state;

    public static void main(String[] args) {
        lock1 = new Lock();
        lock2 = new Lock();
        Process1 p1 = new Process1();
        Process2 p2 = new Process2();
        p1.start();
        p2.start();
    }

    public static void geraException() throws Exception {
        int valor = (int) (Math.random()*10);
        if(valor%2 == 0)
            throw new Exception();
    }
}

class Process1 extends Thread {
    public void run() {
        DeadlockException.state++;
        synchronized (DeadlockException.lock2) {
            synchronized (DeadlockException.lock1) {
                DeadlockException.state++;
            }
        }
    }
}

class Process2 extends Thread {
    public void run() {
        DeadlockException.state++;
        synchronized (DeadlockException.lock1) {
            synchronized (DeadlockException.lock2) {
                DeadlockException.state++;
            }
        }
    }
}
```



```
<?xml version="1.0" encoding="iso-8859-1"?>
<elements>
  <object>
    <classname>name1</classname>
    <name>teste</name>
    <pair>
      <def>23</def>
      <use>24</use>
    </pair>
    <pair>
      <def>25</def>
      <use>37</use>
    </pair>
    <pair>
      <def>23</def>
      <use>100</use>
    </pair>
    <def>1</def>
    <use>2</use>
  </object>
  <object>
    <classname>name2</classname>
    <name>teste2</name>
    <pair>
      <def>24</def>
      <use>23</use>
    </pair>
    <pair>
      <def>26</def>
      <use>25</use>
    </pair>
  </object>
</elements>
```

Java PathfinderX

Saída na Tela + Arquivos XML + Arquivos Threads

```
----- transition #14 thread: 2
gov.nasa.jpfi.jvm.choice.ThreadChoiceFromSet (Thread-0>Thread-1)
DeadlockException.java:58      : synchronized (DeadlockException.lock1) {
----- transition #15 thread: 2
gov.nasa.jpfi.jvm.choice.ThreadChoiceFromSet (Thread-0>Thread-1)
DeadlockException.java:58      : synchronized (DeadlockException.lock1) {
DeadlockException.java:59      : synchronized (DeadlockException.lock2) {
----- snapshot #1
Thread index=1, name=Thread-0, status=BLOCKED, this=Process1@223, priority=5, lockCount=0
owned locks: Lock@219
blocked on: Lock@218
call stack:
  at Process1.run (DeadlockException.java:39)
Thread index=2, name=Thread-1, status=BLOCKED, this=Process2@248, priority=5, lockCount=0
owned locks: Lock@218
blocked on: Lock@219
call stack:
  at Process2.run (DeadlockException.java:59)
----- results
error #1: gov.nasa.jpfi.jvm.NetDeadlockedProperty "deadlock encountered: thread
index=..."
----- search finished: 06/11/07
17:18
```



```
<?xml version="1.0" encoding="iso-8859-1"?>
<elements>
  <object>
    <classname>name1</classname>
    <name>teste</name>
    <pair>
      <def>5</def>
      <use>11</use>
    </pair>
  </object>
  <object>
    <classname>name2</classname>
    <name>teste2</name>
    <pair>
      <def>14</def>
      <use>23</use>
    </pair>
  </object>
  <exception>
    <classname>name1</classname>
    <name>teste3</name>
    <pair>
      <def>18</def>
      <use>36</use>
    </pair>
  </exception>
</elements>
```



```
22 DeadlockException.state++;
23 synchronized (DeadlockException.lock2) {
24   synchronized (DeadlockException.lock1) {
25     DeadlockException.state++;
26     synchronized (DeadlockException.lock1) {
27       synchronized (DeadlockException.lock2) {
28         DeadlockException.state++;
29         synchronized (DeadlockException.lock1) {
30           synchronized (DeadlockException.lock2) {
31             DeadlockException.state++;
32             synchronized (DeadlockException.lock1) {
33               synchronized (DeadlockException.lock2) {
34                 DeadlockException.state++;
35                 synchronized (DeadlockException.lock1) {
36                   synchronized (DeadlockException.lock2) {
37                     DeadlockException.state++;
38                     synchronized (DeadlockException.lock1) {
39                       synchronized (DeadlockException.lock2) {
40                         DeadlockException.state++;
41                         synchronized (DeadlockException.lock1) {
42                           synchronized (DeadlockException.lock2) {
43                             DeadlockException.state++;
44                             synchronized (DeadlockException.lock1) {
45                               synchronized (DeadlockException.lock2) {
46                                 DeadlockException.state++;
47                                 synchronized (DeadlockException.lock1) {
48                                   synchronized (DeadlockException.lock2) {
49                                     DeadlockException.state++;
50                                     synchronized (DeadlockException.lock1) {
51                                       synchronized (DeadlockException.lock2) {
52                                         DeadlockException.state++;
53                                         synchronized (DeadlockException.lock1) {
54                                           synchronized (DeadlockException.lock2) {
55                                             DeadlockException.state++;
56                                             synchronized (DeadlockException.lock1) {
57                                               synchronized (DeadlockException.lock2) {
58                                                 DeadlockException.state++;
59                                                 synchronized (DeadlockException.lock1) {
60                                                   synchronized (DeadlockException.lock2) {
61                                                     DeadlockException.state++;
62                                                     synchronized (DeadlockException.lock1) {
63                                                       synchronized (DeadlockException.lock2) {
64                                                         DeadlockException.state++;
65                                                         synchronized (DeadlockException.lock1) {
66                                                           synchronized (DeadlockException.lock2) {
67                                                             DeadlockException.state++;
68                                                             synchronized (DeadlockException.lock1) {
69                                                               synchronized (DeadlockException.lock2) {
70                                                                 DeadlockException.state++;
71                                                                 synchronized (DeadlockException.lock1) {
72                                                                   synchronized (DeadlockException.lock2) {
73                                                                     DeadlockException.state++;
74                                                                     synchronized (DeadlockException.lock1) {
75                                                                     }
76                                                                 }
77                                                                }
78                                                             }
79                                                         }
80                                                     }
81                                                 }
82                                             }
83                                         }
84                                     }
85                                 }
86                             }
87                         }
88                     }
89                 }
90             }
91         }
92     }
93   }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
```

Exemplo Java PathfinderX

Arquivo XML dos pares cobertos

```
<?xml version="1.0" encoding="iso-8859-1"?>
<elements>
  <object>
    <classname>name1</classname>
    <name>teste</name>
    <pair>
      <def>5</def>
      <use>11</use>
    </pair>
  </object>
  <object>
    <classname>name2</classname>
    <name>teste2</name>
    <pair>
      <def>14</def>
      <use>23</use>
    </pair>
  </object>
  <exception>
    <classname>name1</classname>
    <name>teste3</name>
    <pair>
      <def>18</def>
      <use>36</use>
    </pair>
  </exception>
</elements>
```

Arquivo XML dos pares não cobertos

```
<?xml version="1.0" encoding="iso-8859-1"?>
<elements>
  <object>
    <classname>name1</classname>
    <name>teste</name>
    <pair>
      <def>78</def>
      <use>93</use>
    </pair>
  </object>
  <exception>
    <classname>name1</classname>
    <name>teste3</name>
    <pair>
      <def>21</def>
      <use>42</use>
    </pair>
  </exception>
</elements>
```

Disciplinas Úteis

- MAC0332 – Engenharia de Software
- MAC0340 – Laboratório de Engenharia de Software
- MAC0211 – Laboratório de Programação I
- MAC0242 – Laboratório de Programação II
- MAC0442 – Programação Orientada a Objetos
- MAC0338 – Análise de Algoritmos

Agradecimentos

- Ana Cristina Vieira de Melo
- Kléber da Silva Xavier
- Simone Hanazumi