Fine-grained, accurate and scalable source code differencing

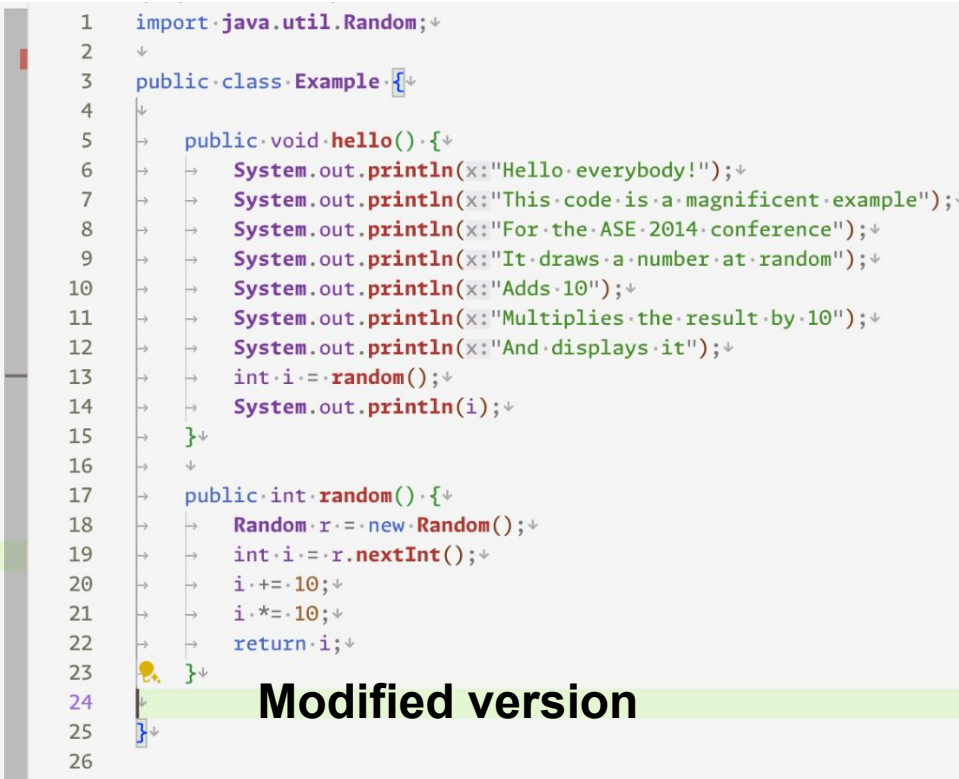
Jean-Rémy Falleri
Bordeaux INP

Matias Martinez
Univ. Poli. Catalunya

Understanding code changes

```java
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example");
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies by 10");
        System.out.println("And displays it");
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        System.out.println(i);
    }

}
```

Original version



```java
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example");
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies the result by 10");
        System.out.println("And displays it");
        int i = random();
        System.out.println(i);
    }

    public int random() {
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        return i;
    }

}
```

Modified version

Textual evolution inference

1 Code model

1. Foo
2. Bar
3. Baz

list of text lines

2 Action model

1. Foo
2. Bar
3. Baz

delete line

1. Foo
2. Bar
3. Baz

add line

3 Min. action sequence from the *original* to *modified* code

1. Foo
2. Bar
3. Baz
4. Bat

original

1. Hello
2. Foo
3. Bar

modified

1. Foo
2. Bar
3. Baz
4. Bat

1. Hello
2. Foo
3. Bar

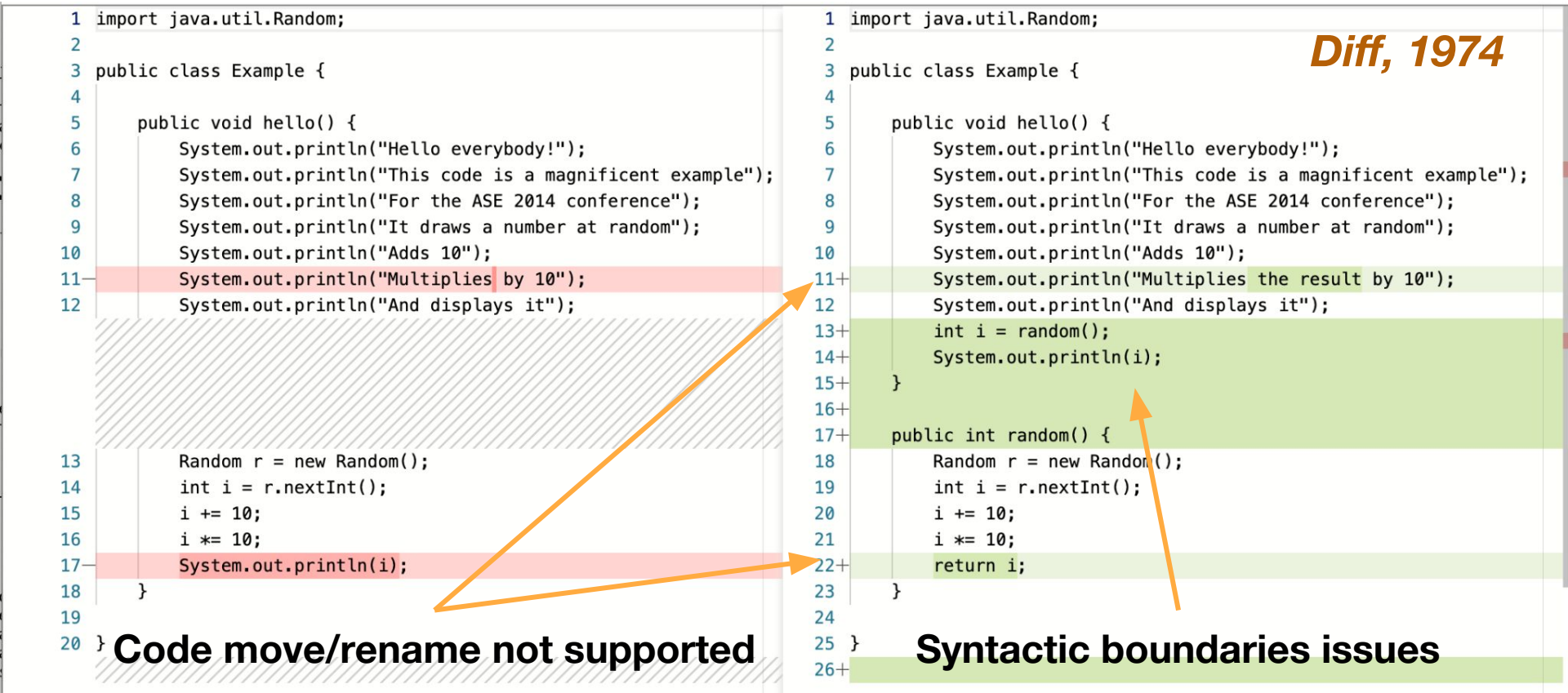
Textual evolution inference

```
1  import java.util.Random;
2
3  public class Example {
4
5      public void hello() {
6          System.out.println("Hello everybody!");
7          System.out.println("This code is a magnificent example");
8          System.out.println("For the ASE 2014 conference");
9          System.out.println("It draws a number at random");
10         System.out.println("Adds 10");
11-        System.out.println("Multiplies by 10");
12         System.out.println("And displays it");
13         Random r = new Random();
14         int i = r.nextInt();
15         i += 10;
16         i *= 10;
17-        System.out.println(i);
18     }
19
20 }
```

```
1  import java.util.Random;
2
3  public class Example {
4
5      public void hello() {
6          System.out.println("Hello everybody!");
7          System.out.println("This code is a magnificent example");
8          System.out.println("For the ASE 2014 conference");
9          System.out.println("It draws a number at random");
10         System.out.println("Adds 10");
11+        System.out.println("Multiplies the result by 10");
12         System.out.println("And displays it");
13+        int i = random();
14+        System.out.println(i);
15+    }
16+
17+    public int random() {
18         Random r = new Random();
19         int i = r.nextInt();
20         i += 10;
21         i *= 10;
22+        return i;
23     }
24
25 }
26+
```

Diff, 1974

Textual evolution inference



Syntactic evolution inference

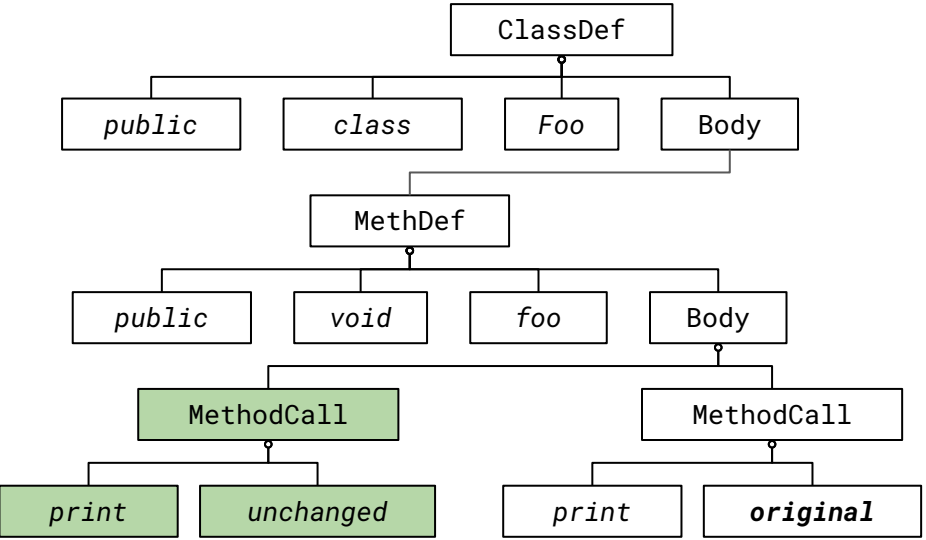
```java
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example");
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies by 10");
        System.out.println("And displays it");
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        System.out.println(i);
    }

}
```

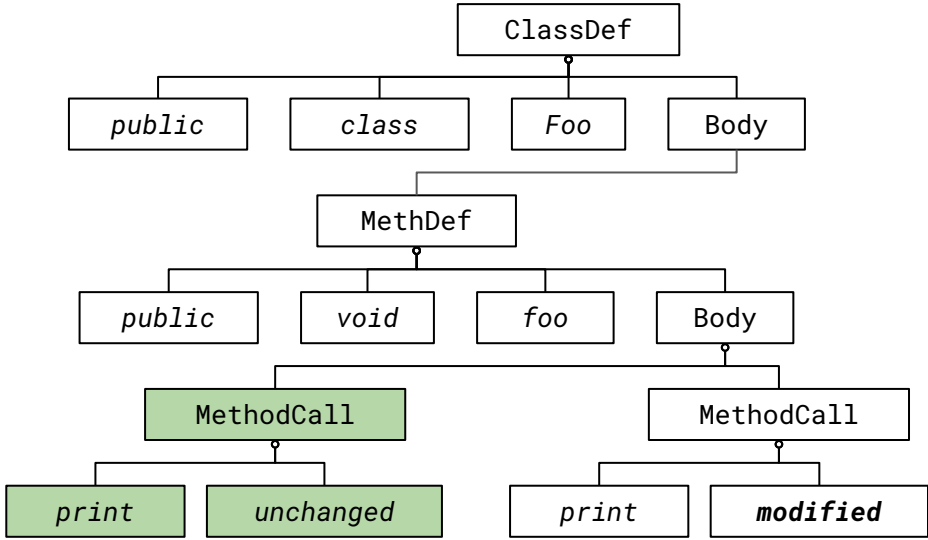
```java
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example");
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies the result by 10");
        System.out.println("And displays it");
        int i = random();
        System.out.println(i);
    }

    public int random() {
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        return i;
    }

}
```

GumTree, 2014

Language-agnostic syntactic differencing approach (around 15 languages supported)

GumTree in a nutshell: top-down phase



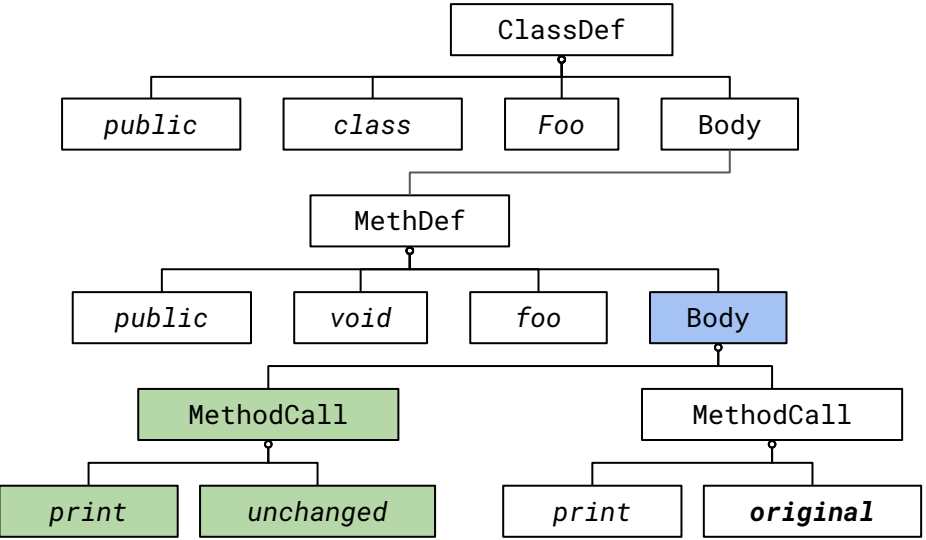


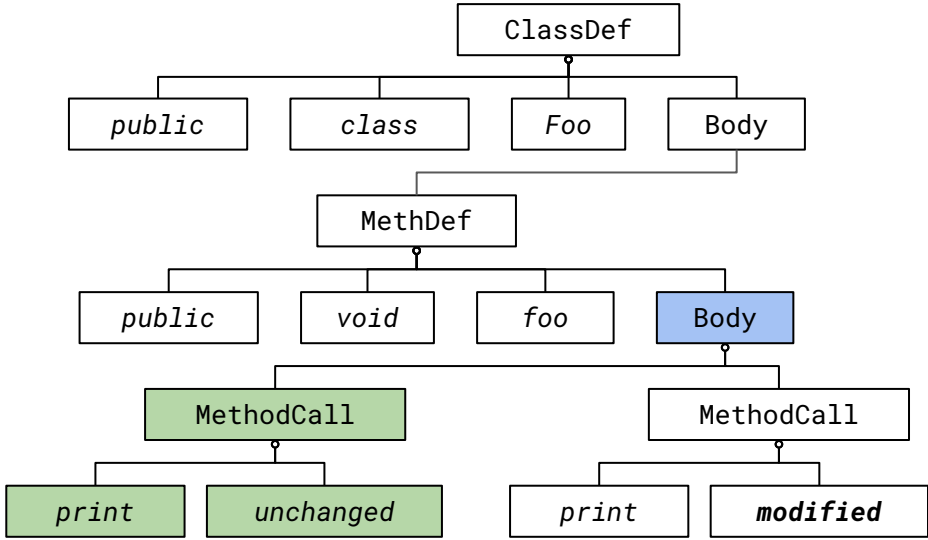

```
public class Foo {
    public void foo() {
      print("unchanged");
      print("original");
    }
}
```

➡️ **min_size threshold**

```
public class Foo {
    public void foo() {
      print("unchanged");
      print("modified");
    }
}
```

GumTree in a nutshell: bottom-up phase

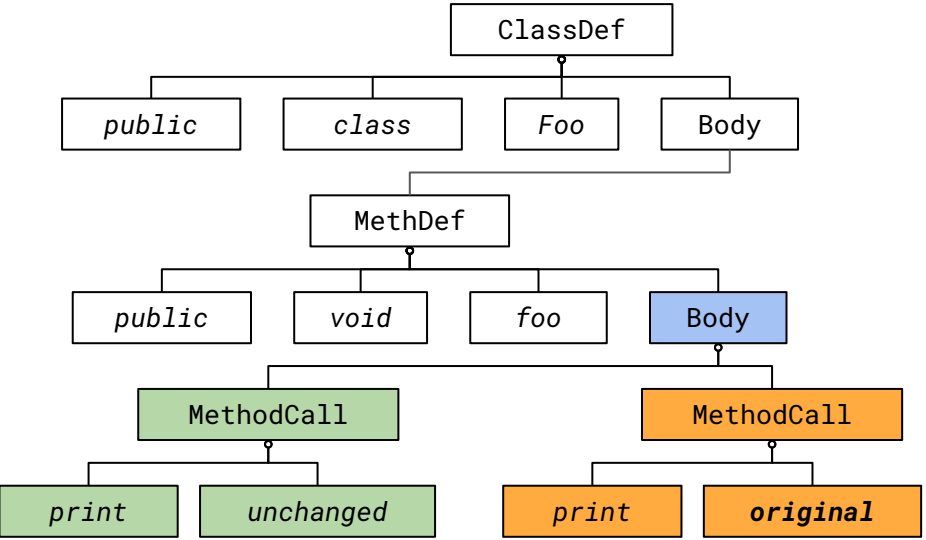


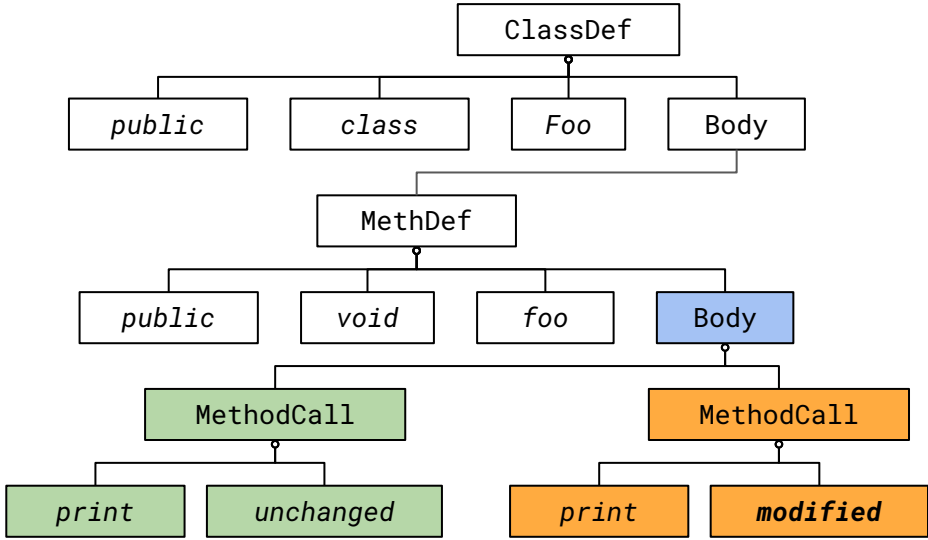



```
public class Foo {
    public void foo() {
      print("unchanged");
      print("original");
    }
}
```

```
public class Foo {
    public void foo() {
      print("unchanged");
      print("modified");
    }
}
```

GumTree in a nutshell: recovery phase



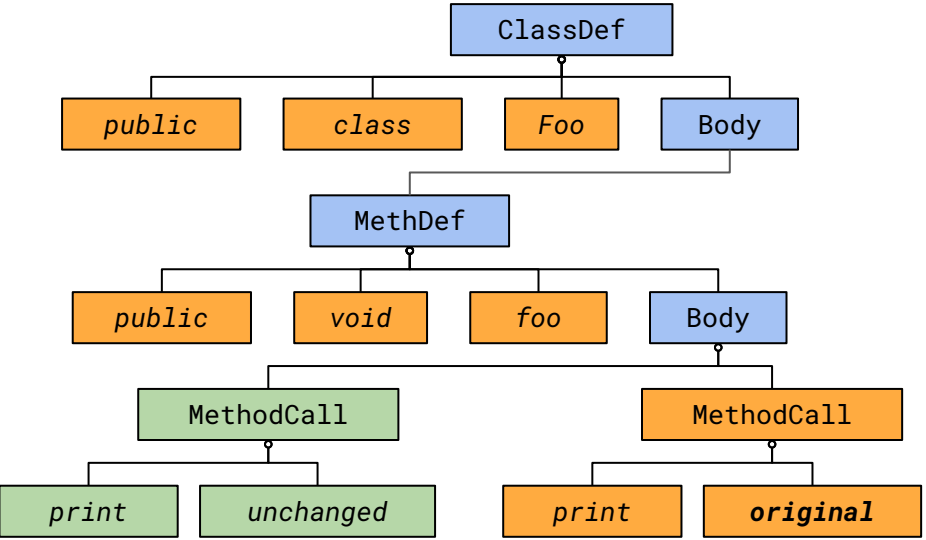


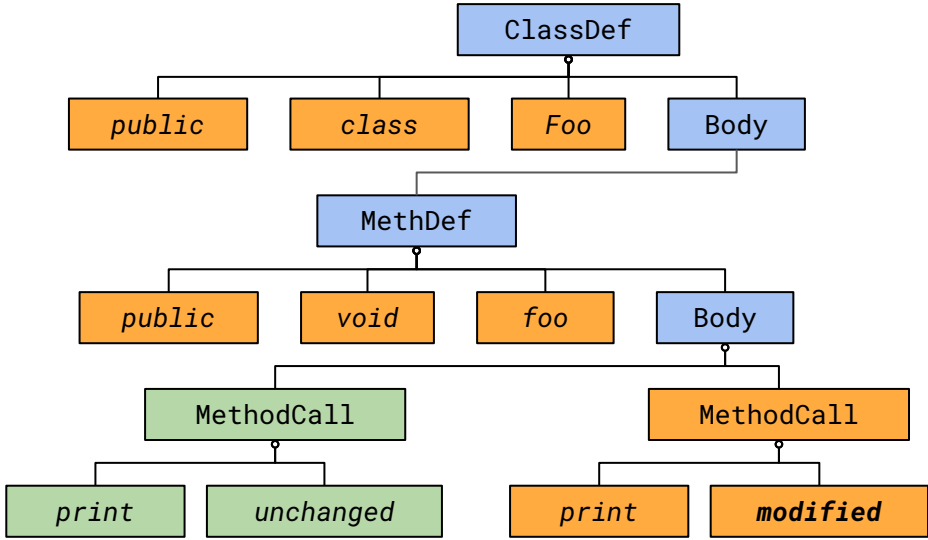

```
public class Foo {
    public void foo() {
      print("unchanged");
      print("original");
    }
}
```

O(n^3) ➡️ max_size threshold

```
public class Foo {
    public void foo() {
      print("unchanged");
      print("modified");
    }
}
```

GumTree in a nutshell: with recovery

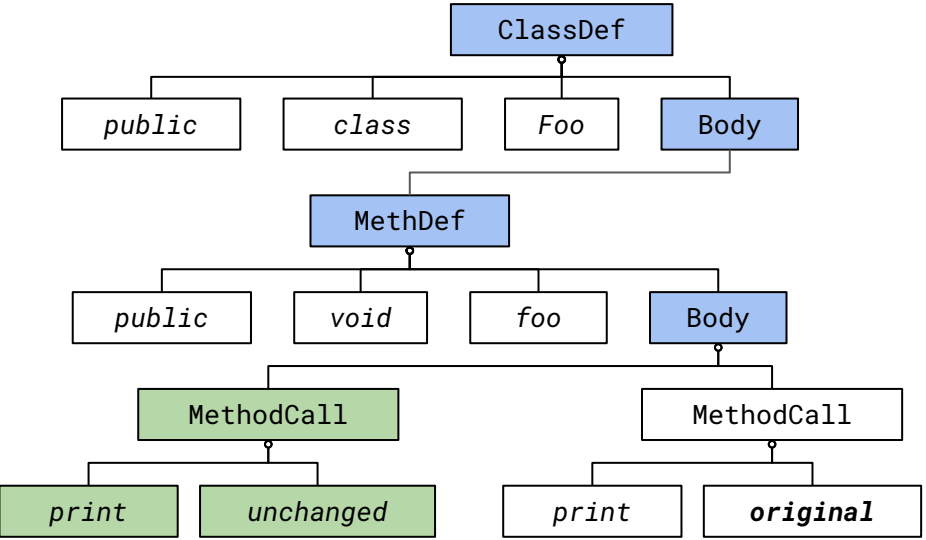


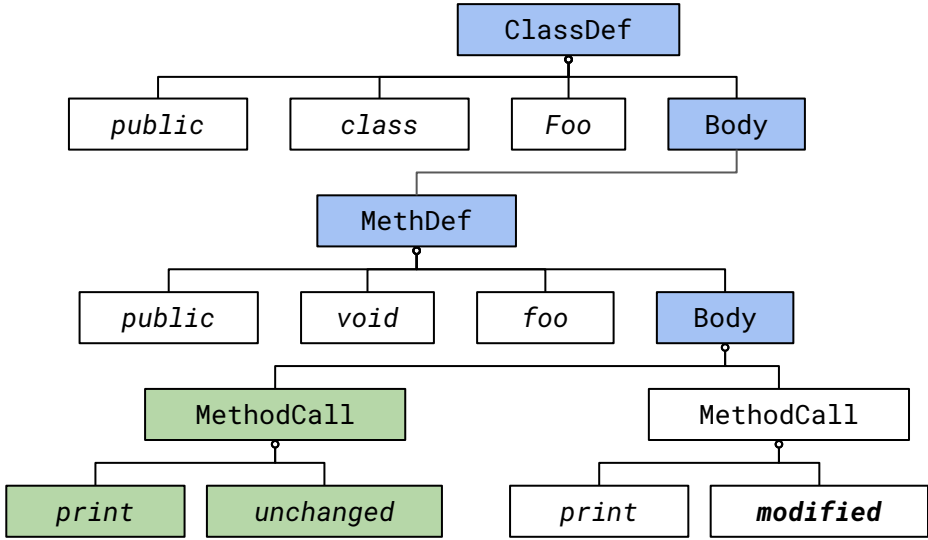



```
public class Foo {
    public void foo() {
      print(“unchanged”);
      print(“original”);
    }
}
```

```
public class Foo {
    public void foo() {
      print(“unchanged”);
      print(“modified”);
    }
}
```

GumTree in a nutshell: without recovery



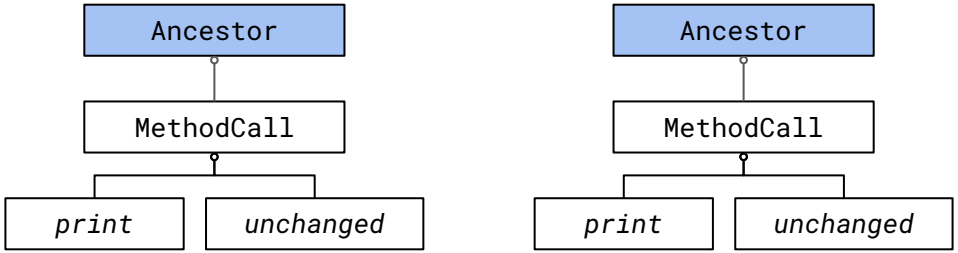



```
public class Foo {
    public void foo() {
      print("unchanged");
      print("original");
    }
}
```

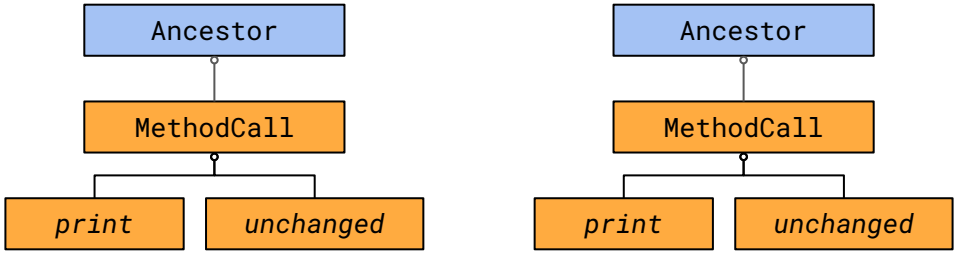
```
public class Foo {
    public void foo() {
      print("unchanged");
      print("modified");
    }
}
```

GumTree simple: a fast recovery heuristic

Simple Recovery: isomorphism



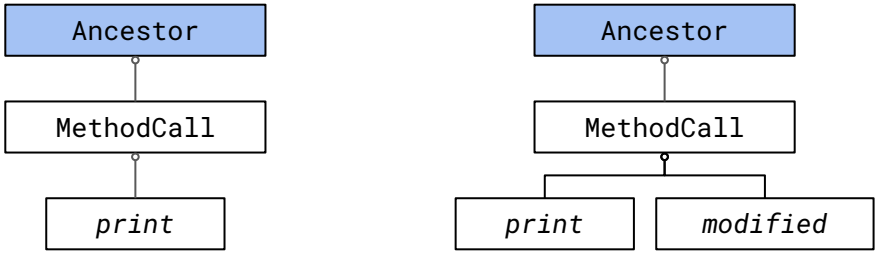
Simple Recovery: isomorphism



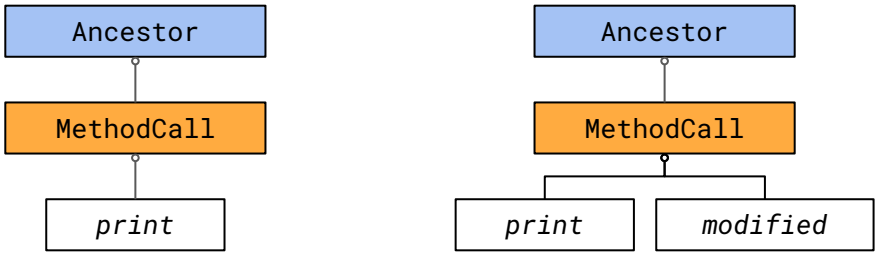
Simple Recovery: iso-structure

Simple Recovery: iso-structure

Simple Recovery: uni-type

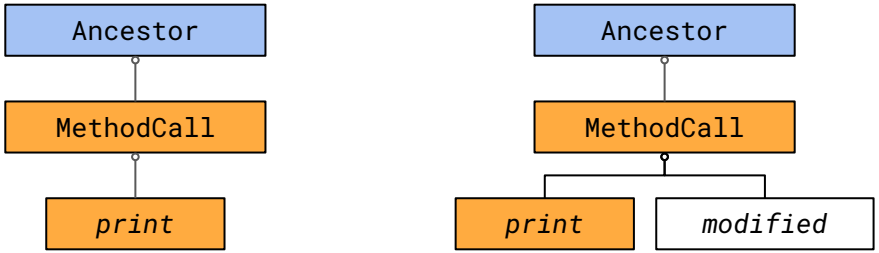


Simple Recovery: uni-type

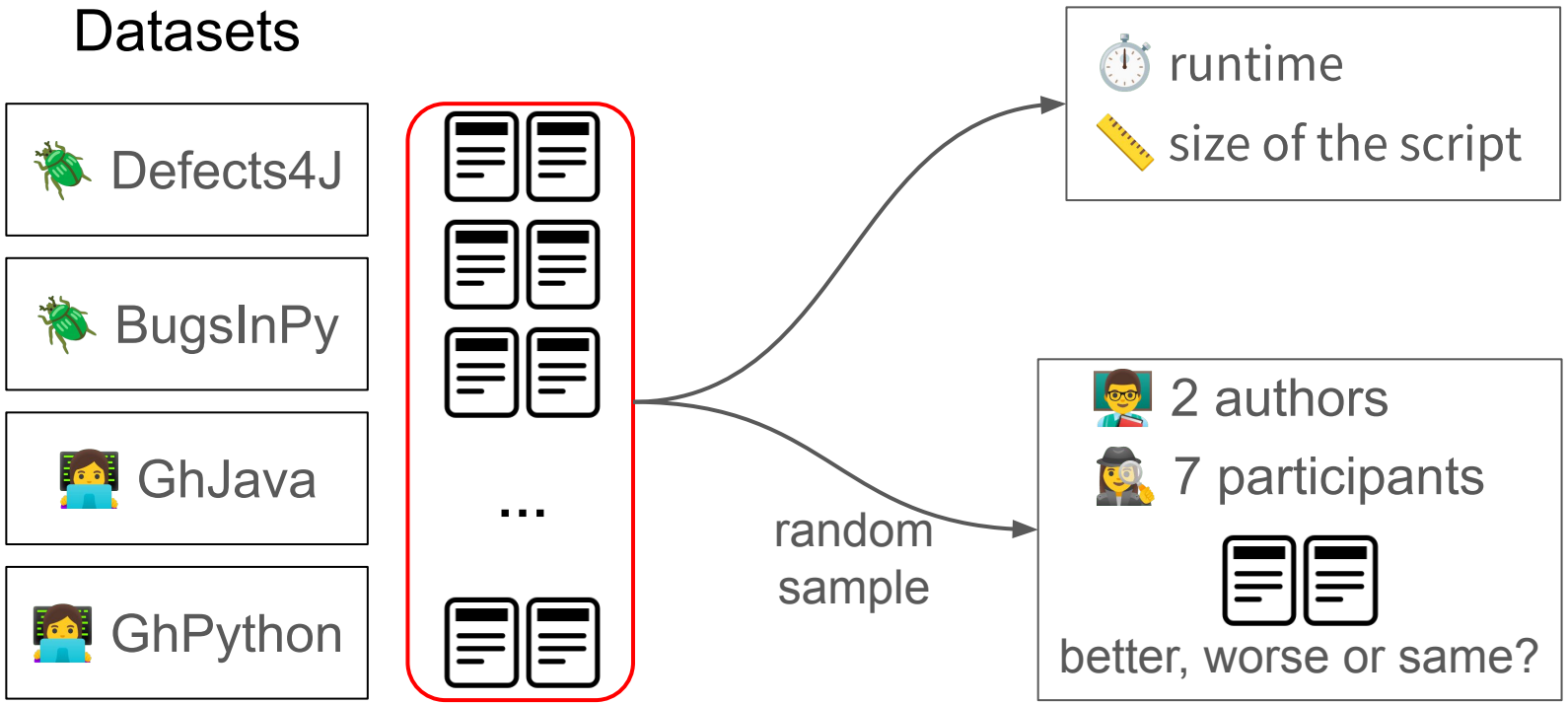


recursive call to gumtree-simple

Simple Recovery: uni-type



Evaluation


Datasets
Defects4J
BugsInPy
GhJava
GhPython
...
random sample
runtime
size of the script
2 authors
7 participants
better, worse or same?

Edit-script size and runtime (quantitative)

Runtime

	simple	
	faster	slower
D4J	0.94	0.06
BIP	0.90	0.10
GHJ	0.93	0.07
GHP	0.89	0.11

Between 50% and 300% matching speedup

Edit-script size

	simple		
	smaller	same	bigger
D4J	0.73	0.22	0.05
BIP	0.74	0.21	0.05
GHJ	0.37	0.50	0.14
GHP	0.41	0.51	0.07

Around 50% smaller median edit-script size

Diff perception (qualitative)

	authors			participants		
	better	mixed	worse	better	mixed	worse
BIP	23	1	1	24	1	0
D4J	22	3	0	23	1	1
GHJ	20	2	3	22	1	2
GHP	18	4	3	22	0	3
total	83	10	7	91	3	6

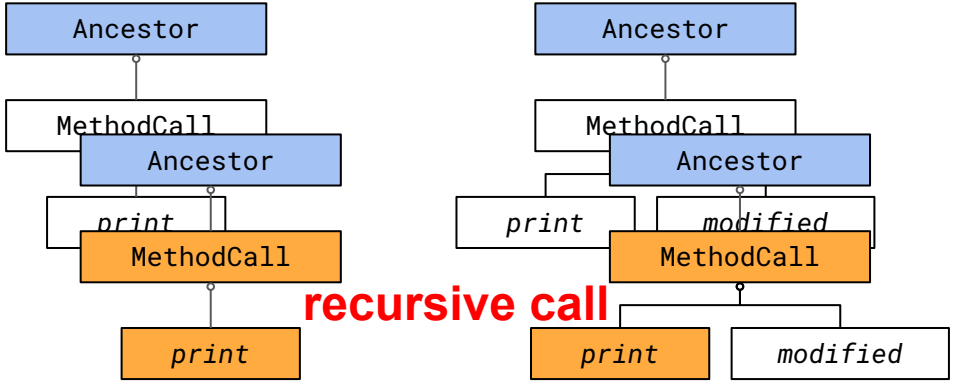
90% of smaller edit-scripts judged better by our participants

```java
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example");
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies by 10");
        System.out.println("And displays it");
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        System.out.println(i);
    }
}
```

Original version

```java
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example");
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies the result by 10");
        System.out.println("And displays it");
        int i = random();
        System.out.println(i);
    }

    public int random() {
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        return i;
    }
}
```

Modified version



	simple		
	smaller	same	bigger
D4J			
BIP			
GHJ			
GHP			

	simple	
	faster	slower
D4J		
BIP		
GH		
GH		

	authors			participants		
	better	mixed	worse	better	mixed	worse
BIP	23	1	1	24	1	0
D4J	22	3	0	23	1	1
GHJ	20	2	3	22	1	2
GHP	18	4	3	22	0	3
total	83	10	7	91	3	6