LaBRI

Présentation scientifique

Jean-Rémy Falleri

Analyse de différences entre fichiers de code source

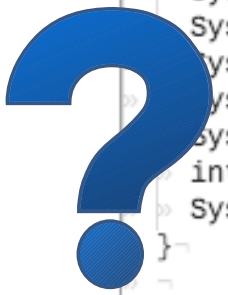
cnrs

université de BORDEAUX

Bordeaux INP AQUITAINE

Context

```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);
  }

}
```

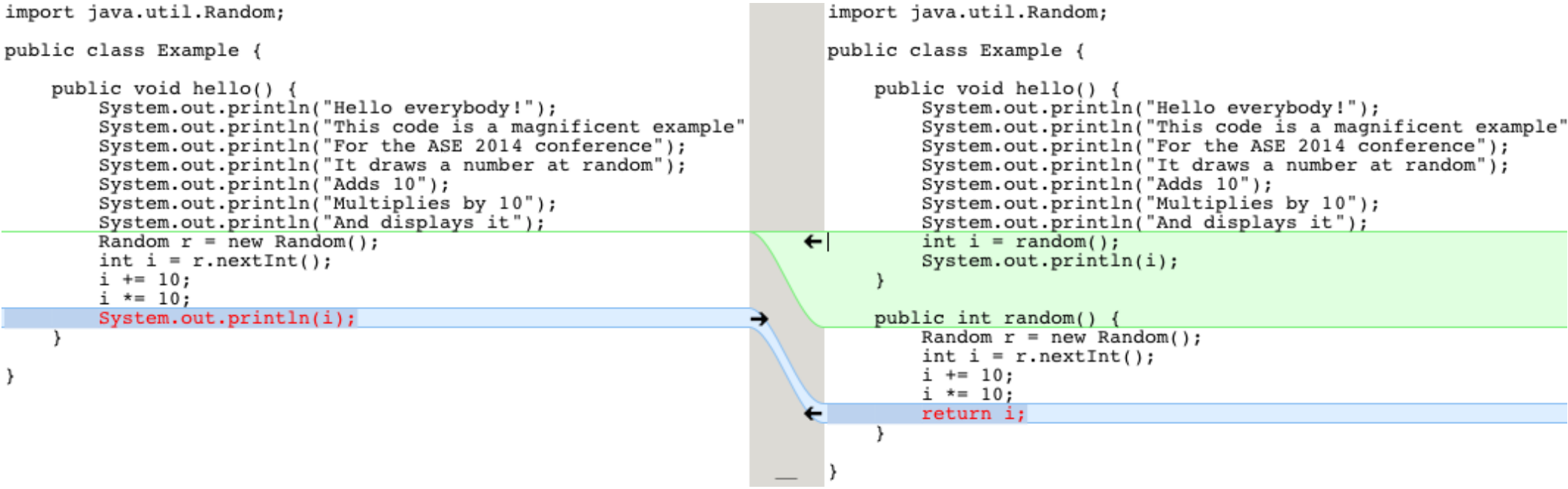


```java
import java.util.Random;

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 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;
  }

}
```

Previous version

Current version

Current approach

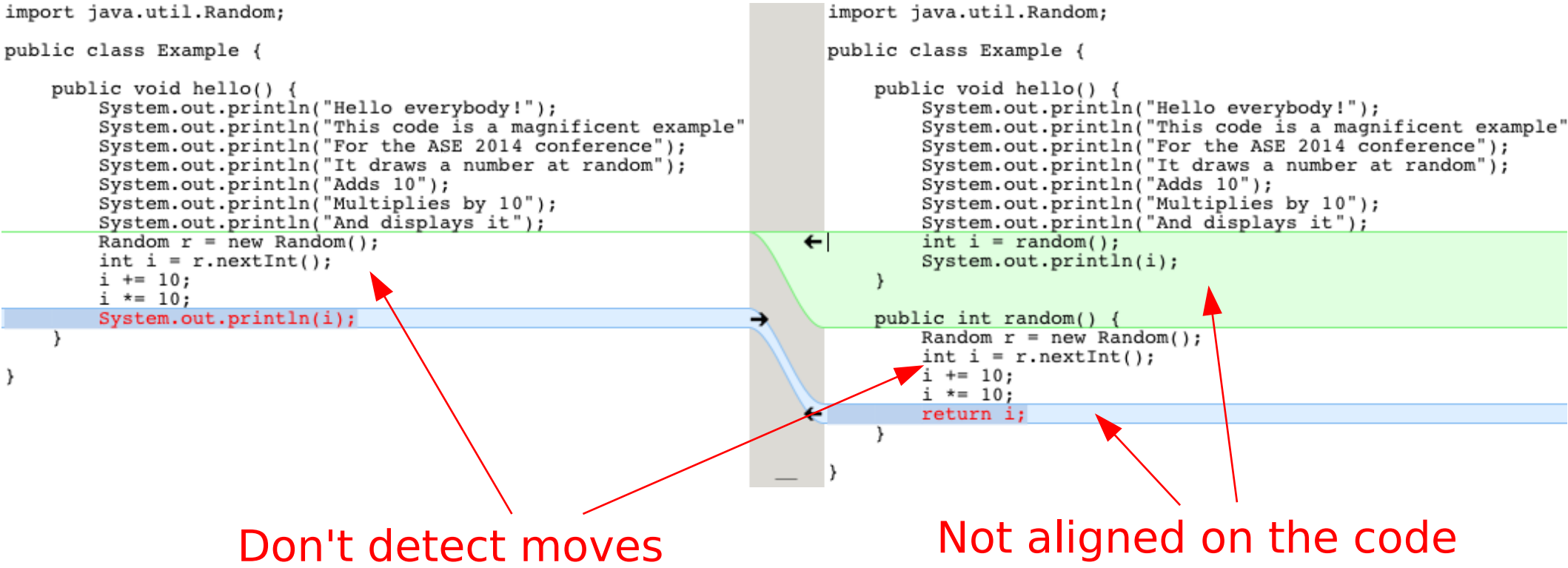


```
import java.util.Random;

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);
    }

}
```

```
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");
        int i = random();
        System.out.println(i);
    }

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

}
```

Meld v1.8

Current approach



Don't represent the developer intent

Our proposal

```
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);
        }

}
```

```
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");
                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

- Differencing algorithm
  - Work on syntax trees
  - Detect code moves
- Tool
  - Efficient
  - Empirically validated
  - Freely available

State of the art

- On text
  - Never aligned on the code structure
- On syntax trees
  - Optimal without moves: *O(n^3)* [27]
  - Optimal with moves: *NP-hard* [4]
  - Very few existing heuristics (ex: [6, 13])

GumTree process

1. Parse code files to code agnostic tree structure
2. **Find mappings between nodes**
3. Deduce code edition actions (as in [6])
4. Output code differences

Finding mappings

1. Top-down
   - Find biggest chunks of unmodified code
2. Bottom-up
   - Propagate mappings to the containers of these chunks (functions, classes, …)
   - Extend mappings in the left-over code of these containers

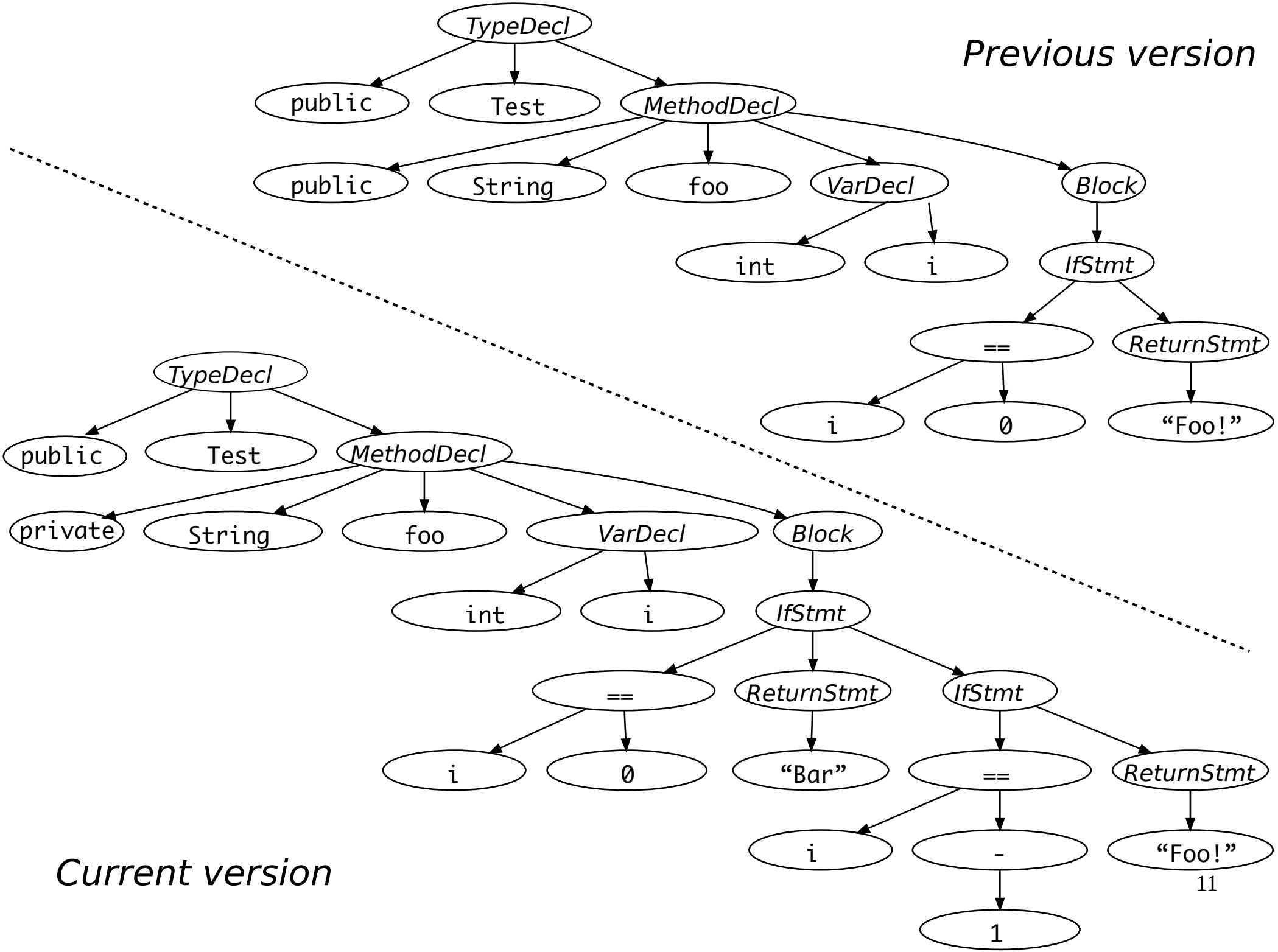
Example

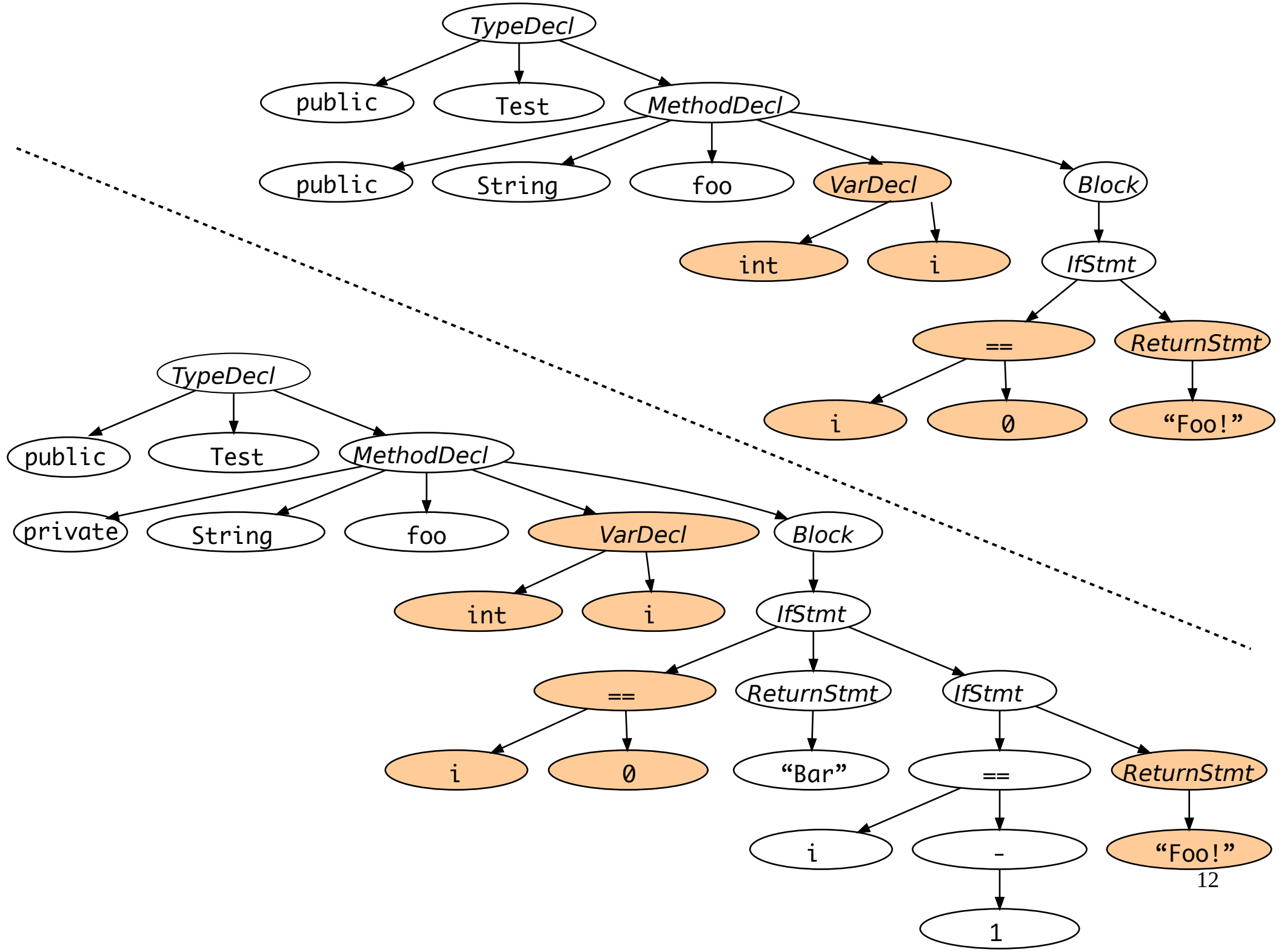
```java
public class Test {
  public String foo(int i) {
    if (i == 0)
      return "Foo!";
  }
}
```

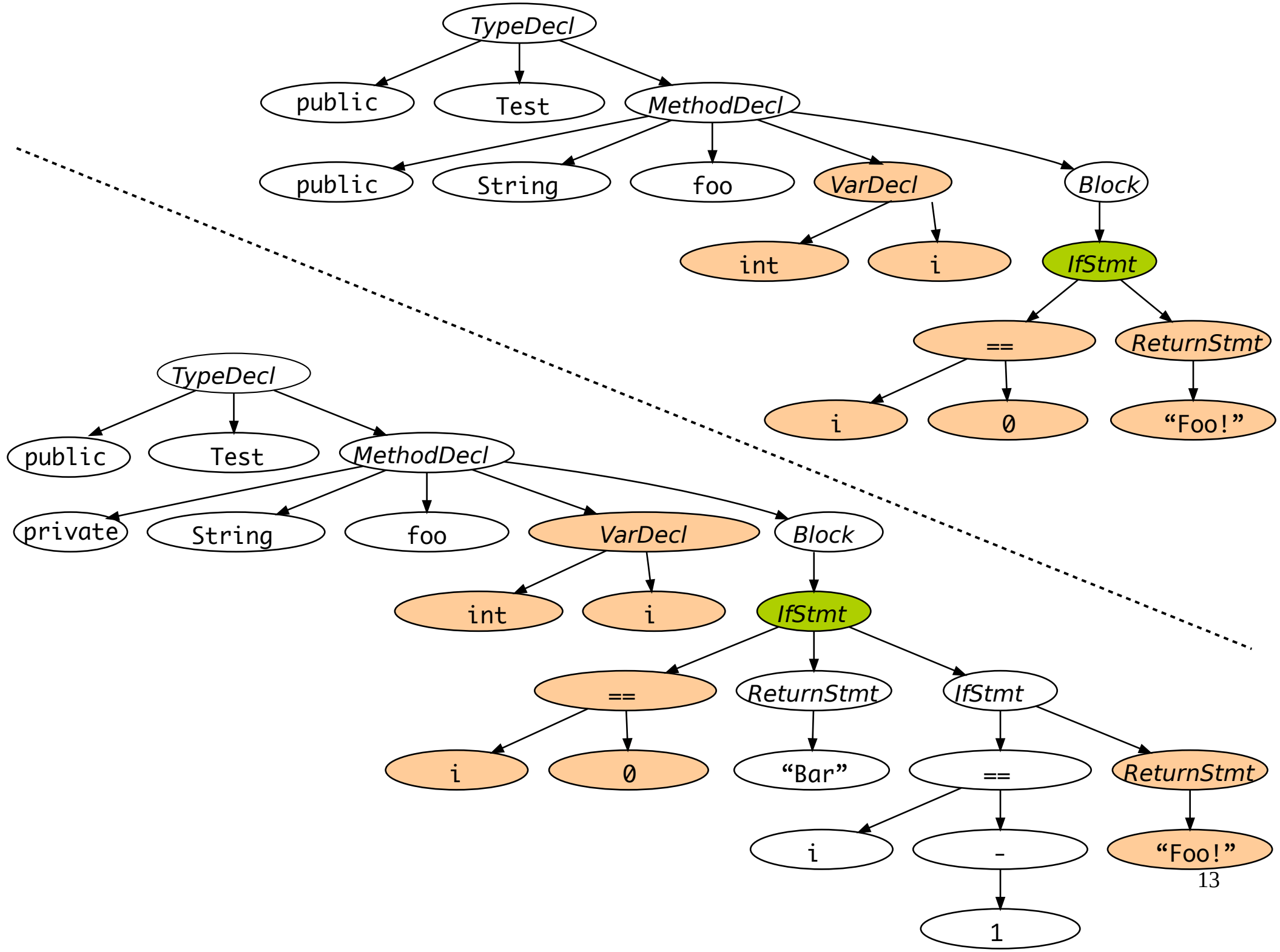
Previous version

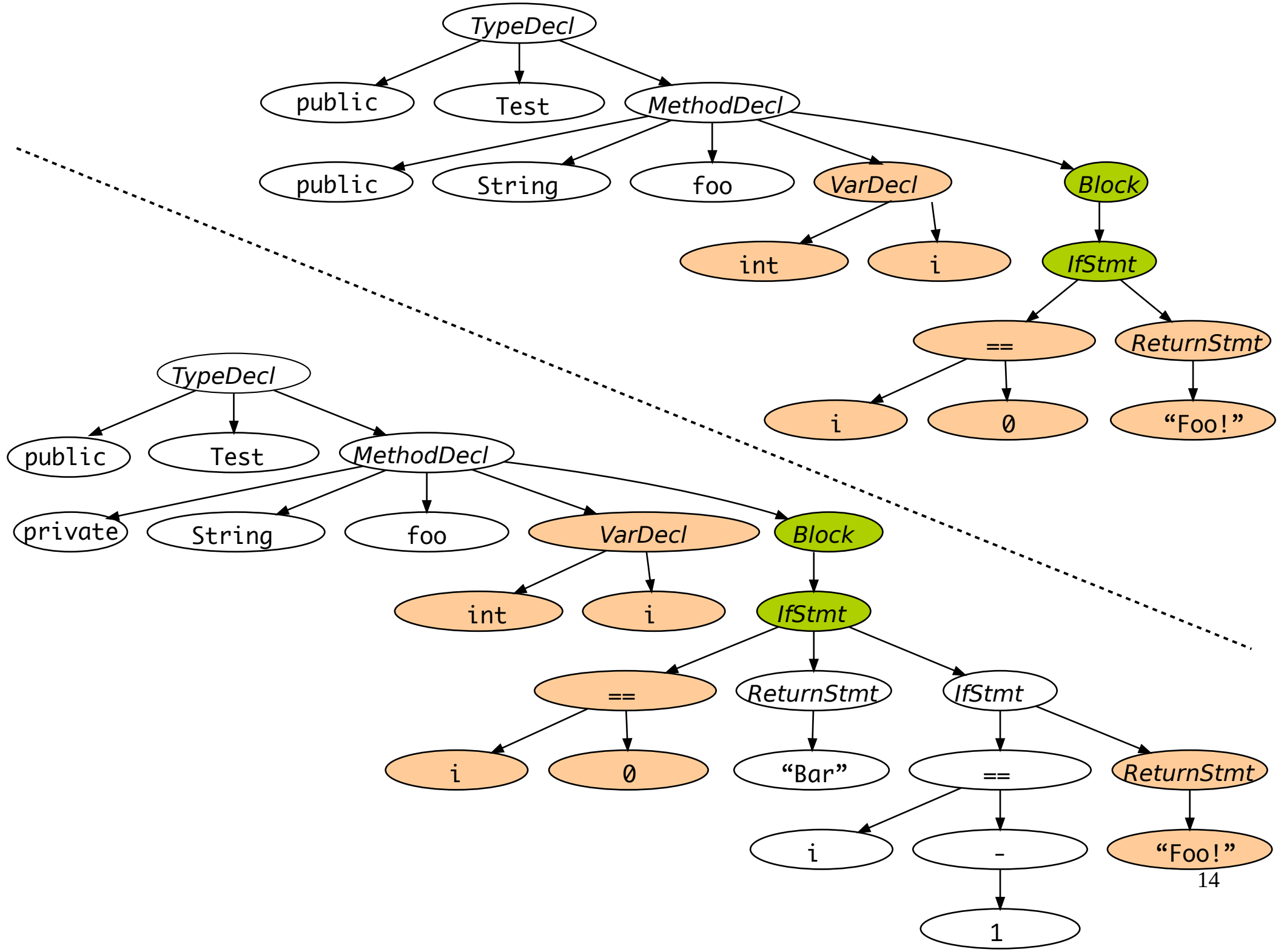
```java
public class Test {
  private String foo(int i) {
    if (i == 0)
      return "Bar";
    else if (i == -1)
      return "Foo!";
  }
}
```

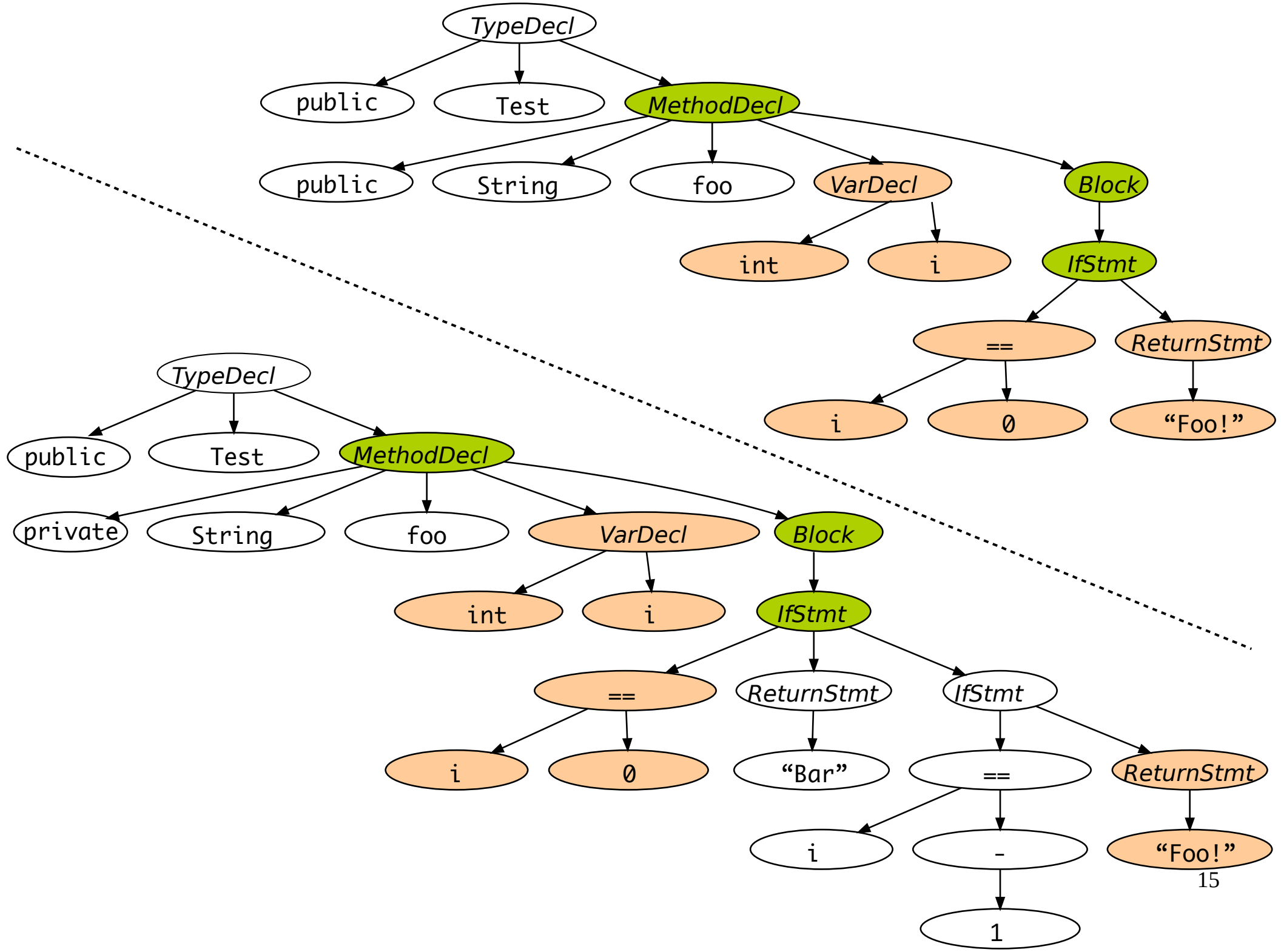
Current version

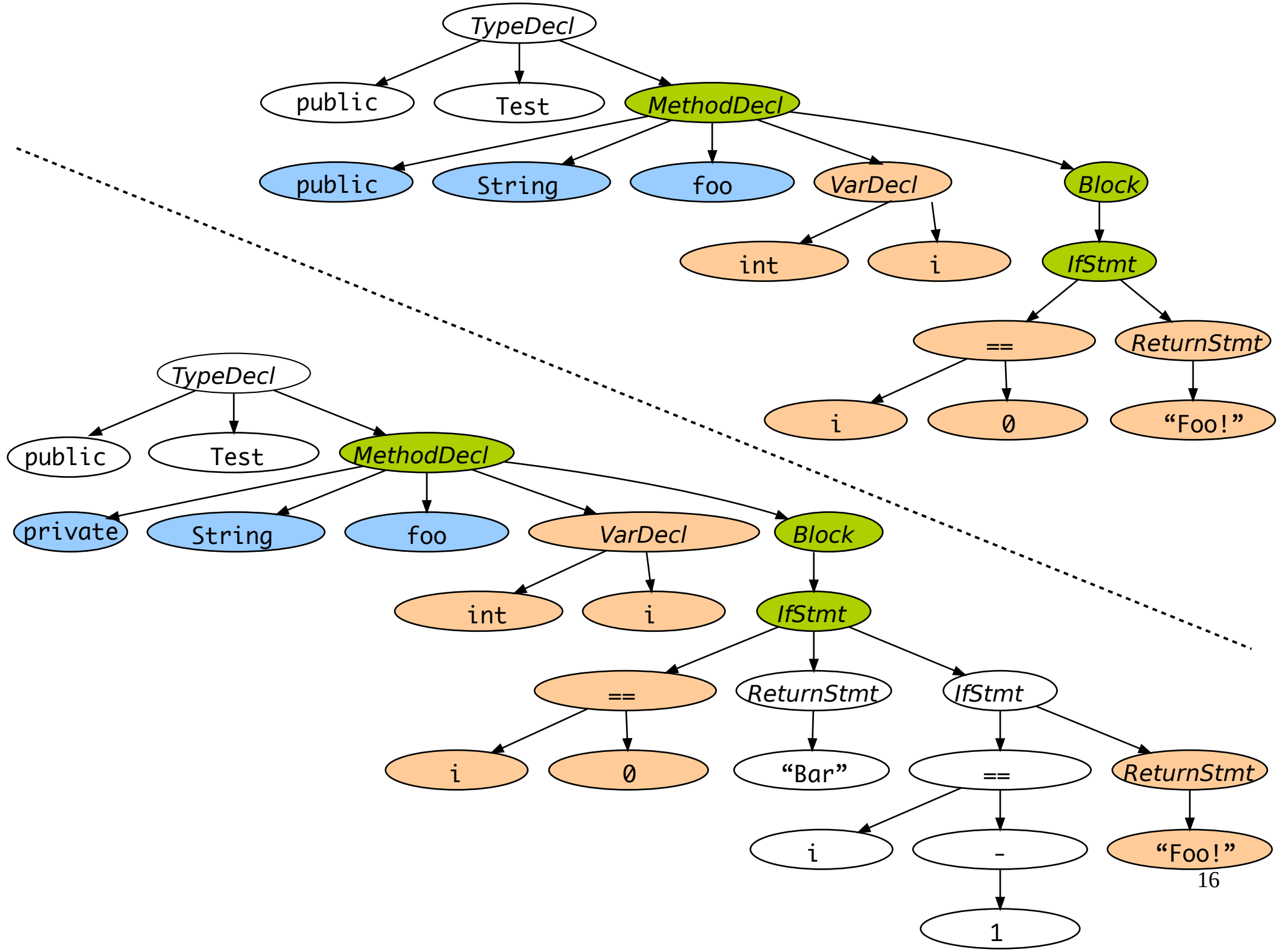

Previous version
TypeDecl
public
Test
MethodDecl
public
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
i
0
“Foo!”
TypeDecl
public
Test
MethodDecl
private
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
IfStmt
i
0
“Bar”
==
ReturnStmt
i
-
“Foo!”
1
Current version

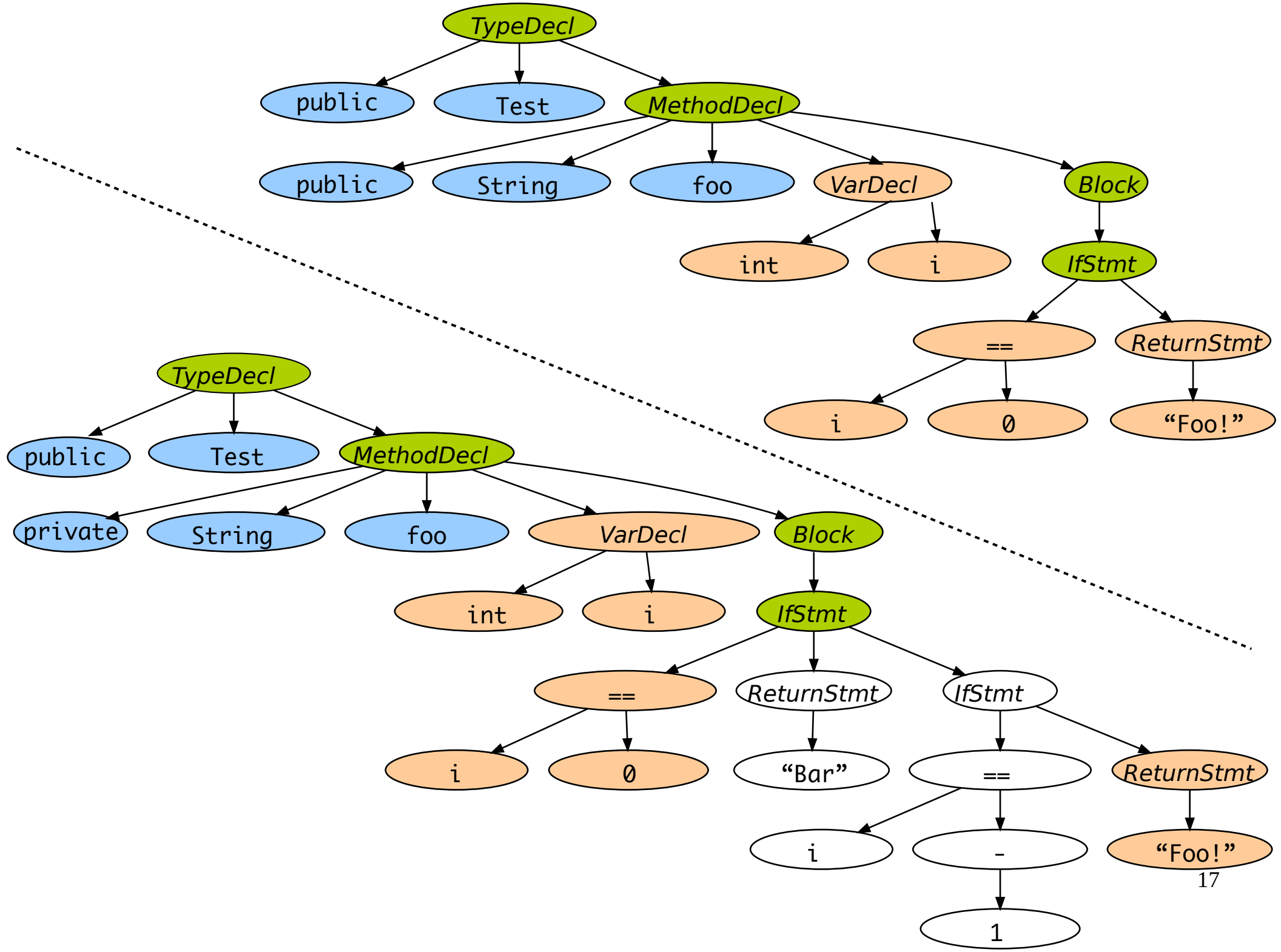

TypeDecl
public
Test
MethodDecl
public
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
i
0
“Foo!”
TypeDecl
public
Test
MethodDecl
private
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
IfStmt
i
0
“Bar”
==
ReturnStmt
i
-
“Foo!”
1


TypeDecl
public
Test
MethodDecl
public
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
i
0
“Foo!”
TypeDecl
public
Test
MethodDecl
private
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
IfStmt
i
0
“Bar”
==
ReturnStmt
i
-
“Foo!”
1


TypeDecl
public
Test
MethodDecl
public
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
i
0
“Foo!”
TypeDecl
public
Test
MethodDecl
private
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
IfStmt
i
0
“Bar”
==
ReturnStmt
i
-
“Foo!”
1


TypeDecl
public
Test
MethodDecl
public
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
i
0
“Foo!”
TypeDecl
public
Test
MethodDecl
private
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
IfStmt
i
0
“Bar”
==
ReturnStmt
i
-
“Foo!”
1


TypeDecl
public
Test
MethodDecl
public
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
i
0
“Foo!”
TypeDecl
public
Test
MethodDecl
private
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
IfStmt
i
0
“Bar”
==
ReturnStmt
i
-
“Foo!”
1


TypeDecl
public
Test
MethodDecl
public
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
i
0
“Foo!”
TypeDecl
public
Test
MethodDecl
private
String
foo
VarDecl
Block
int
i
IfStmt
==
ReturnStmt
IfStmt
i
0
“Bar”
==
ReturnStmt
i
-
“Foo!”
1

Output

```
public class Test {
  public String foo(int i) {
    if (i == 0)
      return "Foo!";
  }
}
```

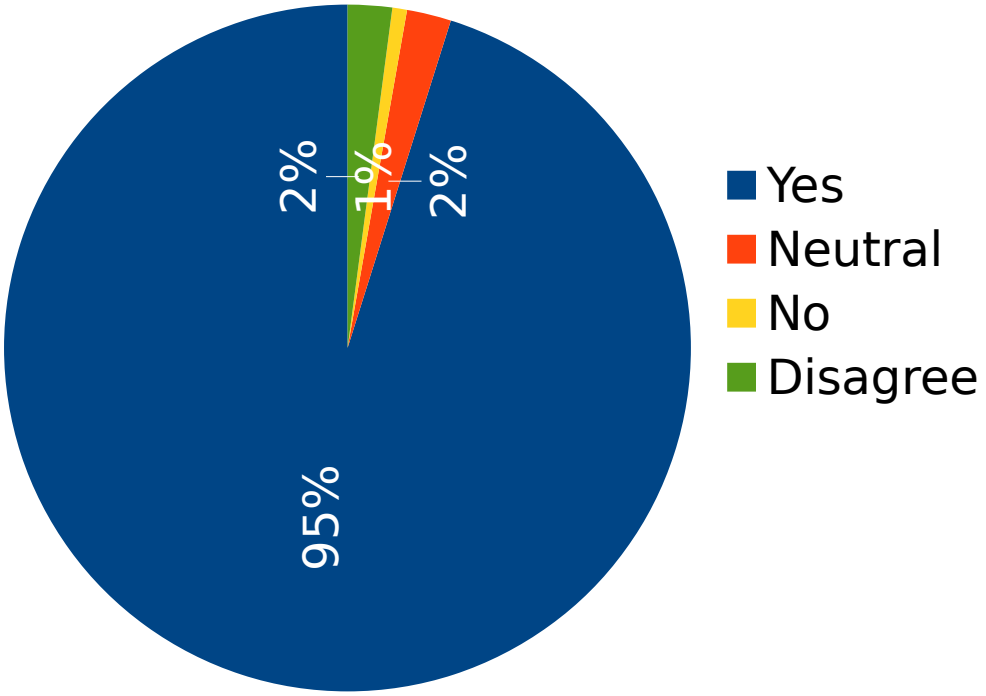
Previous state

```
public class Test {
  private String foo(int i) {
    if (i == 0)
      return "Bar";
    else if (i == -1)
      return "Foo!";
  }
}
```

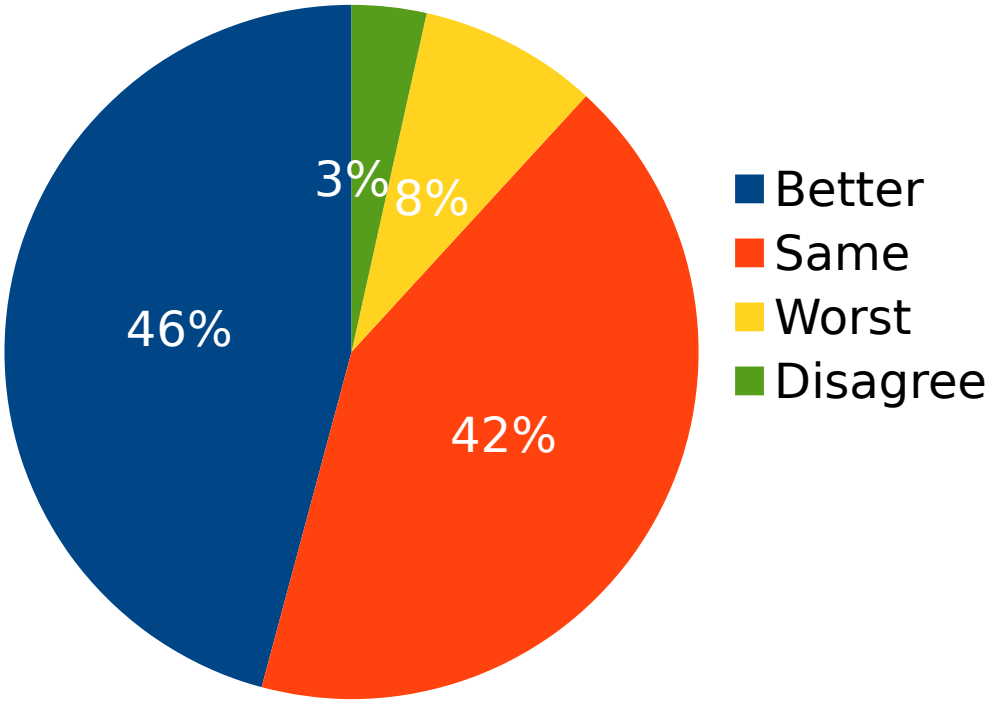
Current state

Human evaluation

GumTree does a good job?



GumTree vs text diff?



Conclusion

- Differencing algorithm
  - Detect code moves
  - Work on syntax trees
- Tool
  - Efficient
    - O(n^2)
    - Only 2 or 3 times slower than parsing
  - Empirically validated
    - Shorter edit script with more move than other heuristics
  - Freely available
    - Two articles