Introduction to Compiler Construction in a Java World

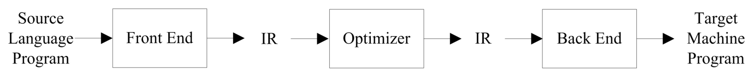
Bill Campbell, Swami Iyer, Bahar Akbal-Delibaş

Errata

Here you can find a listing of known errors in our text. If you find others, please let us know about them at `j--@cs.umb.edu`. We appreciate your feedback.

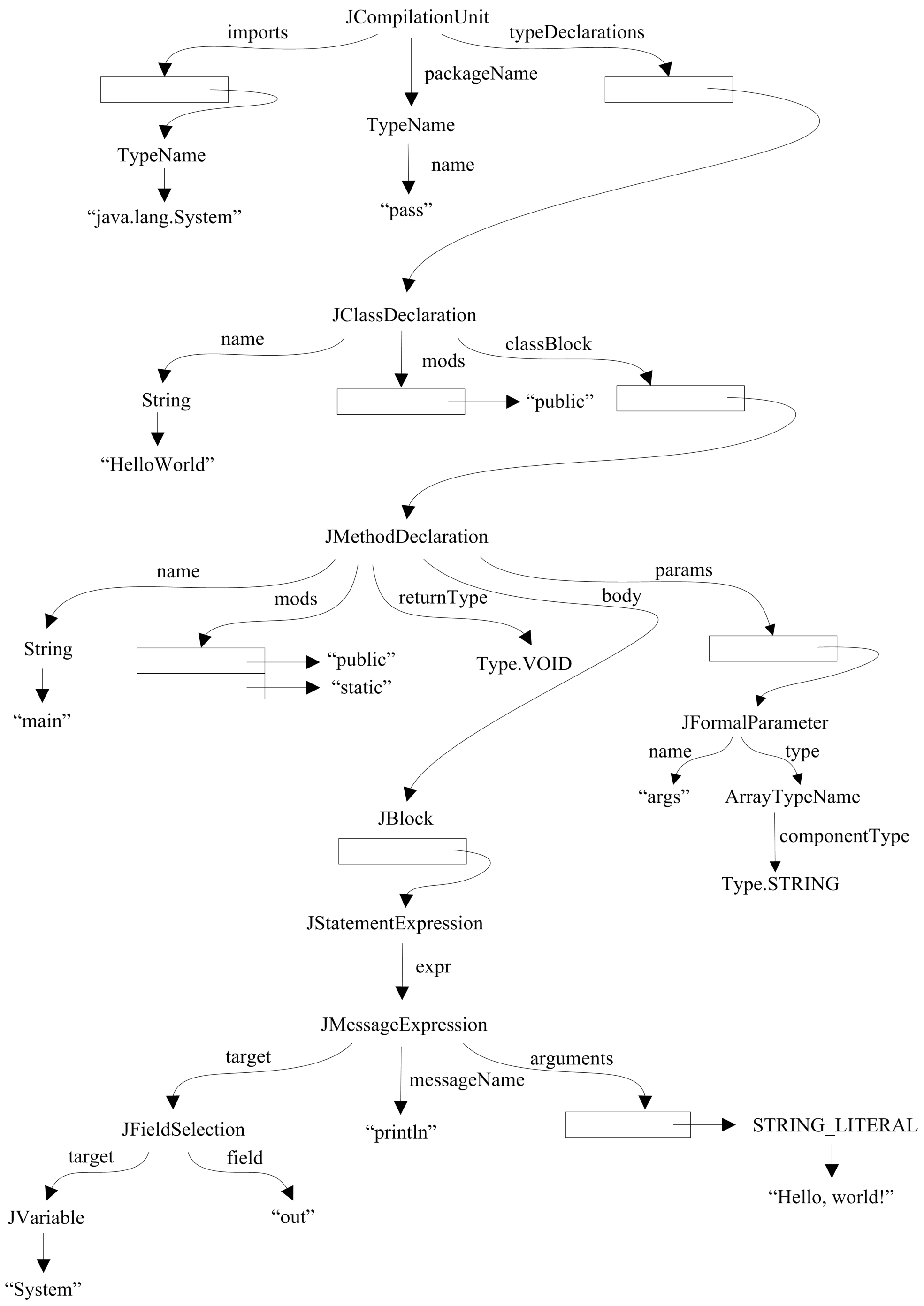
Chapter 1: Compilation

On page 6, Figure 1.6 the "JVM Code" on the left of the figure must be "Source Language Program". Here is the updated figure.



Reported by Antoine Marchal on Jun 17, 2014

On page 14, Figure 1.9 is missing the formal parameters of the `main()` method. Here is the updated figure.



Reported by Bill Campbell on Jan 28, 2013

On page 18, the following line

For example, the *j*-- program ...

should be

For example, the *j*-- program ...

Reported by Pierre Schaus on Feb 7, 2013

On page 19, the following code snippet

```
public class DivisionTest extends TestCase {
    ...
```

should be

```
package junit;

import junit.framework.TestCase;
import pass.Division;

public class DivisionTest extends TestCase {
    ...
```

Reported by Daisuke Tanaka on Feb 1, 2013

Chapter 2: Lexical Analysis

On page 41, the set of moves M currently given by

$$M = \{m(0,a) = 1, m(0,b) = 1, m(1,a) = 1, m(1,b) = 1, m(1,\epsilon) = 0, m(1,b) = 2\}$$

should be

$$M = \{m(0,a) = 1, m(1,a) = 1, m(1,b) = 1, m(1,\epsilon) = 0, m(1,b) = 2\}$$

Reported by Pierre Schaus on Feb 11, 2013

On page 46, the following line in Definition 2.6

... set of states S includes s and ...

should be

... set of states S includes S and ...

Reported by Pierre Schaus on Feb 7, 2013

Chapter 3: Parsing

On page 59, the following code snippet

```
package pass;f
    ...
```

should be

```
package pass;
    ...
```

Reported by Pierre Schaus on Feb 14, 2013

On page 88, the following line in Algorithm 3.6

where $X_j ::= \beta_1|\beta_2|\ldots|\beta_k$ are the current rules defining X_i

should be

where $X_j ::= \beta_1|\beta_2|\ldots|\beta_k$ are the current rules defining X_j

Reported by Pierre Schaus on Feb 14, 2013

Chapter 5: JVM Code Generation

The tables on page 187 and 189 are incorrect. Here are the updated tables.

	x	a[i]	o.f	C.sf
lhs = y	iload y' [dup] istore x'	aload a' iload i' iload y' [dup_x2] iastore	aload o' iload y [dup_x1] putfield f	iload y' [dup] putstatic sf
lhs += y	iload x' iload y' iadd [dup] istore x'	aload a' iload i' dup2 iaload iload y' iadd [dup_x2] iastore	aload o' dup getfield f iload y' iadd [dup_x1] putfield f	getstatic sf iload y' iadd [dup] putstatic sf
++lhs	iinc x',1 [iload x']	aload a' iload i' dup2 iaload iconst_1 iadd [dup_x2] iastore	aload o' dup getfield f iconst_1 iadd [dup_x1] putfield f	getstatic sf iconst_1 iadd [dup] putstatic sf
lhs--	[iload x'] iinc x',-1	aload a' iload i' dup2 iaload [dup_x2] iconst_1 isub iastore	aload o' dup getfield f [dup_x1] iconst_1 isub putfield f	getstatic sf [dup] iconst_1 isub putstatic sf

	x	a[i]	o.f	C.sf

`codegenLoadLhsLvalue()`	`[none]`	`aload a'`
		`iload i'`
			`aload o'`	`[none]`

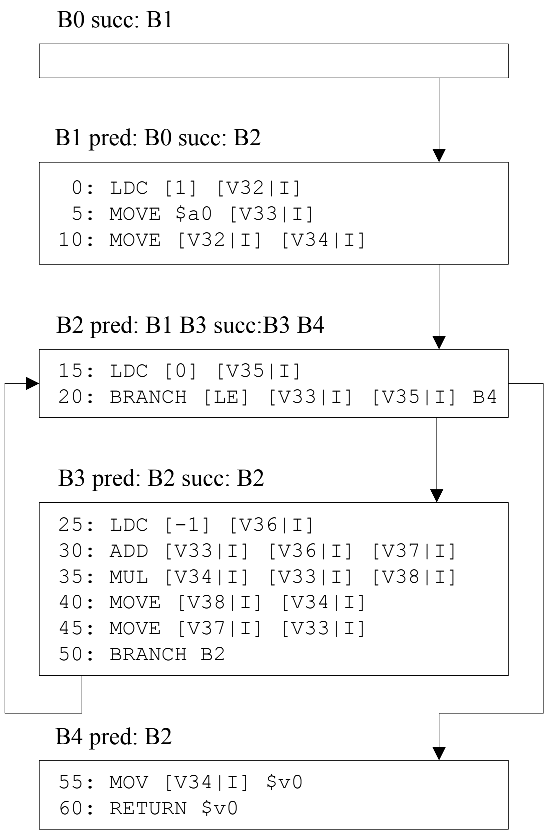
`codegenLoadLhsRvalue()`	`iload x'`	`dup2`
		`iaload`
			`dup`
			`getfield f`
				`getstatic sf`

`codegenDuplicateRvalue()`	`dup`	`dup_x2`	`dup_x1`	`dup`
`codegenStore()`	`istore x'`	`iastore`	`putfield f`	`putstatic sf`

Reported by Bill Campbell on April 15, 2013

Chapter 7: Register Allocation

In Figure 7.1 on page 247 and Figure 7.3 on page 250, the link from block B2 to block B3 is missing. Here are the updated figures.



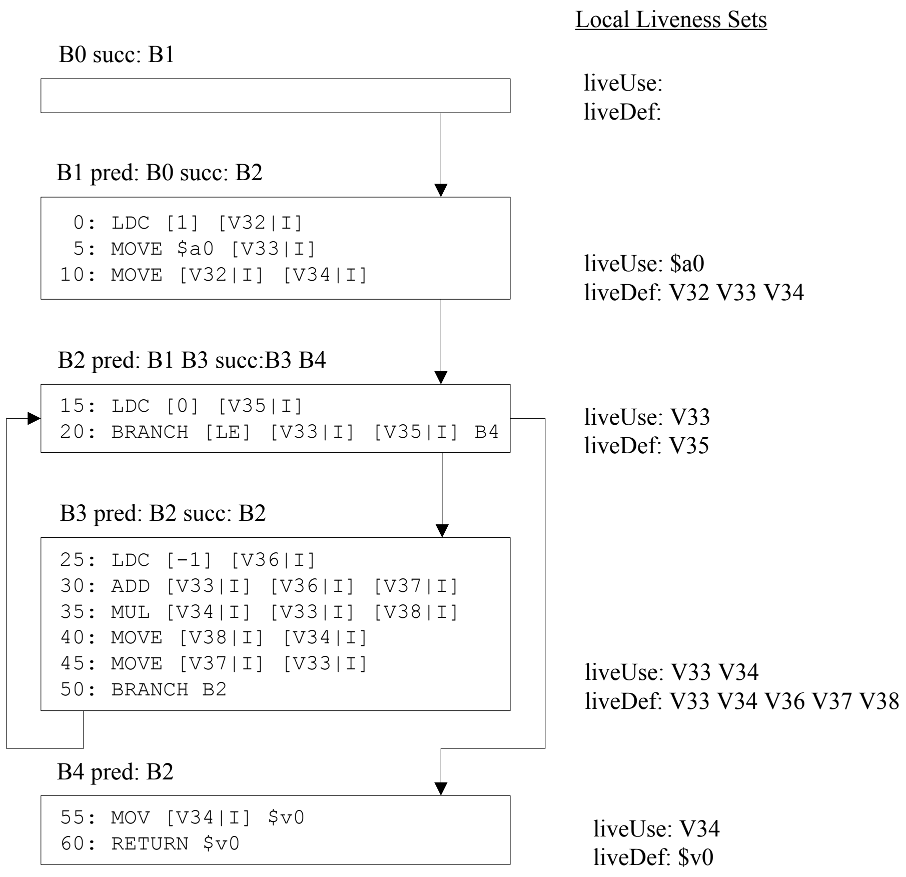
```
B0 succ: B1

B1 pred: B0 succ: B2
 0: LDC [1] [V32|I]
 5: MOVE $a0 [V33|I]
10: MOVE [V32|I] [V34|I]

B2 pred: B1 B3 succ:B3 B4
15: LDC [0] [V35|I]
20: BRANCH [LE] [V33|I] [V35|I] B4

B3 pred: B2 succ: B2
25: LDC [-1] [V36|I]
30: ADD [V33|I] [V36|I] [V37|I]
35: MUL [V34|I] [V33|I] [V38|I]
40: MOVE [V38|I] [V34|I]
45: MOVE [V37|I] [V33|I]
50: BRANCH B2

B4 pred: B2
55: MOV [V34|I] $v0
60: RETURN $v0
```



```
 0: LDC [1] [V32|I]
 5: MOVE $a0 [V33|I]
10: MOVE [V32|I] [V34|I]
```

```
15: LDC [0] [V35|I]
20: BRANCH [LE] [V33|I] [V35|I] B4
```

```
25: LDC [-1] [V36|I]
30: ADD [V33|I] [V36|I] [V37|I]
35: MUL [V34|I] [V33|I] [V38|I]
40: MOVE [V38|I] [V34|I]
45: MOVE [V37|I] [V33|I]
50: BRANCH B2
```

```
55: MOV [V34|I] $v0
60: RETURN $v0
```

Reported by Josef Joller on Dec 15, 2013

Algorithm 7.9 uses `!` for logical not, which is confusing since `!` also represents logical not in *j*--. Here is the corrected version, with `!` replaced by **not**.

Algorithm 1 Graph Coloring Register Allocation

```
Input: The control-flow graph $g$ for a method with LIR that makes use of virtual registers
Output: The same $g$ but with virtual registers replaced by physical registers
  $registersAssignedSuccessfully \leftarrow$ false
  repeat
    repeat
      buildIntervals()
      buildInterferenceGraph()
    until not coalesceRegistersSuccessful()
    buildAdjacencyLists()
    computeSpillCosts()
    pruneGraph()
    $registersAssignedSuccessfully \leftarrow$ assignRegisters()
    if not $registersAssignedSuccessfully$ then
      generateSpillCode()
    end if
  until $registersAssignedSuccessfully$
```

Reported by Bill Campbell on Nov 30, 2014