Exercise 9: Language Extension: Special While Loop

Extend MicroJava's while statement so that a loop can have multiple loop headers and multiple loop bodies, for example:

```
while (n % 2 == 0)     // loop header 1
   n = n / 2;
|| (n % 3 == 0)        // loop header 2
   n = n - 1;
```

The first loop header begins with while, the others with ||. The loop conditions are checked sequentially. If any of them is true, the corresponding statement is executed; after this statement, the program jumps back to the beginning of the loop. The loop terminates if none of the loop conditions is true. The context-free grammar for this looks as follows:

```
WhileStat = "while" "("Condition ")" Statement { "||" "(" Condition ")" Statement } .
```

Turn this grammar into an attributed grammar so that the correct jump instructions are generated. To do this, use the helper methods for labels and jumps described in Section 6.7 (Code.jump(), Code.fJump(), and label.here()). Note that Condition returns a Cond operand.

Solution

```
WhileStat                    (.  Label top = new Label();
                                 Operand x; .)
=  "while"                   (. top.here(); .)
   "(" Condition <↑x> ")"    (.  Code.fJump(x);
                                 x.tLabel.here(); .)
   Statement                 (.  Code.jump(top);
                                 x.fLabel.here(); .)

   {  "||"
      "(" Condition <↑x> ")" (.  Code.fJump(x);
                                 x.tLabel.here(); .)
      Statement              (.  Code.jump(top);
                                 x.fLabel.here(); .)
   } .
```

Condition returns a Cond operand. The subsequent *false jump* leads to the end of the respective loop body, where either the loop ends or the next loop header is checked. As Condition can be a compound Boolean expression, there may be unresolved *true jumps*. Therefore, after each *false jump*, the possibly existing list of *true jumps* is resolved (x.tLabel.here()). At the end of each loop body, the program jumps back to the beginning of the loop.