王建楠 2019013296

实验内容

新增了节点 ForStmt 和 ContinueStmt 分别处理 for/do-while 循环和 continue。

在这此处我将 for/do-while 统一归纳为 ForStmt。

在语义解析时，新增了非终结符 ForExpr 用于处理 for 循环中缺省判断表达式和更新更新表达式的情况，如下：

```
ForStmt     : FOR LPAREN ForExpr SEMICOLON ForExpr SEMICOLON ForExpr RPAREN
Stmt
                { $$ = new ast::ForStmt($3, $5, $7, $9, POS(@1)); }
            | FOR LPAREN DeclStmt ForExpr SEMICOLON ForExpr RPAREN Stmt
                { $$ = new ast::ForStmt($3, $4, $6, $8, POS(@1)); }
            | DO Stmt WHILE LPAREN Expr RPAREN SEMICOLON
                { $$ = new ast::ForStmt($5, $2, POS(@1)); }
            ;

ForExpr     : /* empty */
                { $$ = NULL; }
            | Expr
                { $$ = $1; }
```

ForStmt 中有四个比较重要的成员变量 `init`，`condition`，`update`，`first_condition`，`loop_body`，分别表示初始语句、循环条件、更新语句、第一次条件判断语句（do-while 时 `first_condition = NULL`，for 时 `first_condition=condition`）、循环体。

生成三地址码时，有三个部分，依次为初始语句 & 进入循环前的第一次条件判断、循环体、循环更新语句 & 条件判断语句，用到了三个标签，分别位于循环体的开始位置、循环更新后、条件判断后（即结束位置）。

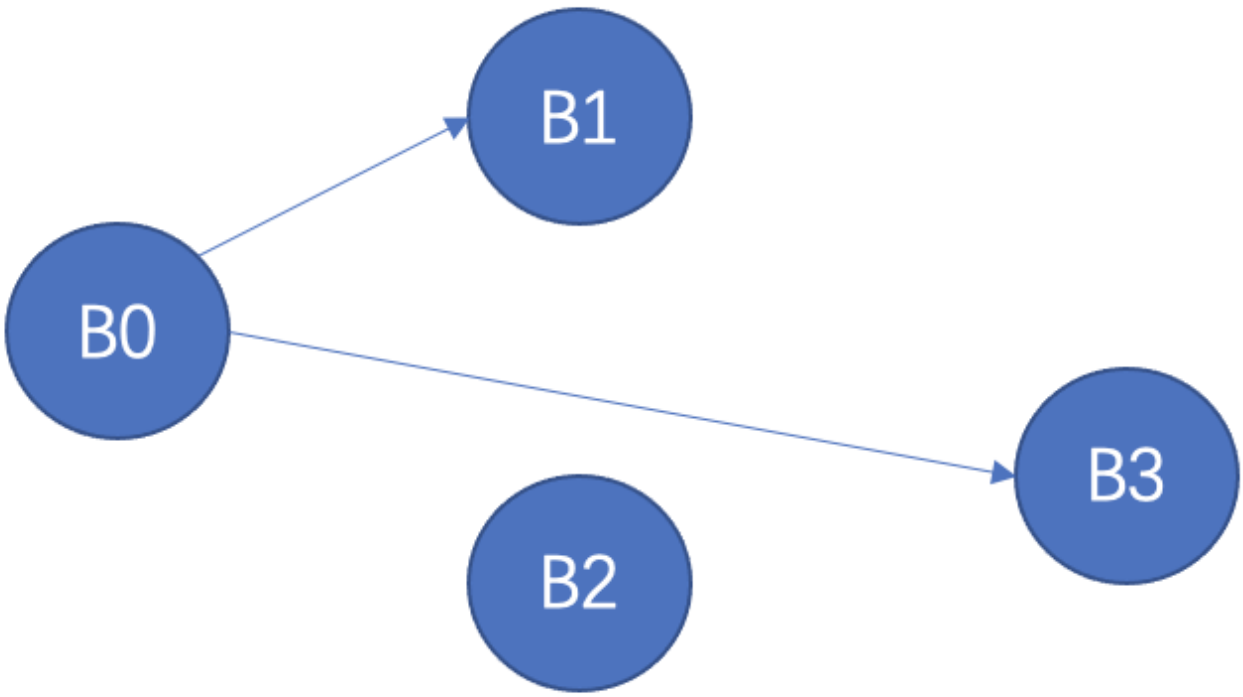
当遇到 break 和 continue 时 `genMarkLebel` 到相应标签位置即可。

思考题

Step 7

```
_main:
        T1 <- 2
        T0 <- T1
        T2 <- 3
        T3 <- (T0 < T2)
        if (T3 == 0) jump __L1
        T5 <- 3
        T4 <- T5
        return T4
        return T0
__L1:
        T6 <- 0
        return T6
```

生成的三地址码有 4 个基本块，从上到下依次为 B0,B1,B2,B3。



Step 8

从执行的指令的条数这个角度第二种翻译方式更好?

以如下代码为例

```
for(int i = 0; i < n; ++i){
    a += i;
}
```

第一种需要执行 $(n+1)$ cond + n body + n br + $(n+1)$ beqz；

第二种需要执行 $(n+1)$ cond + n body + n bnez + 1 beqz

第二种执行代码少了一次 n 次 br，故更好。