

lab挑战实验：短路求值

学号：19373075

姓名：许天识

思路方法

采用实验指导中的控制流法

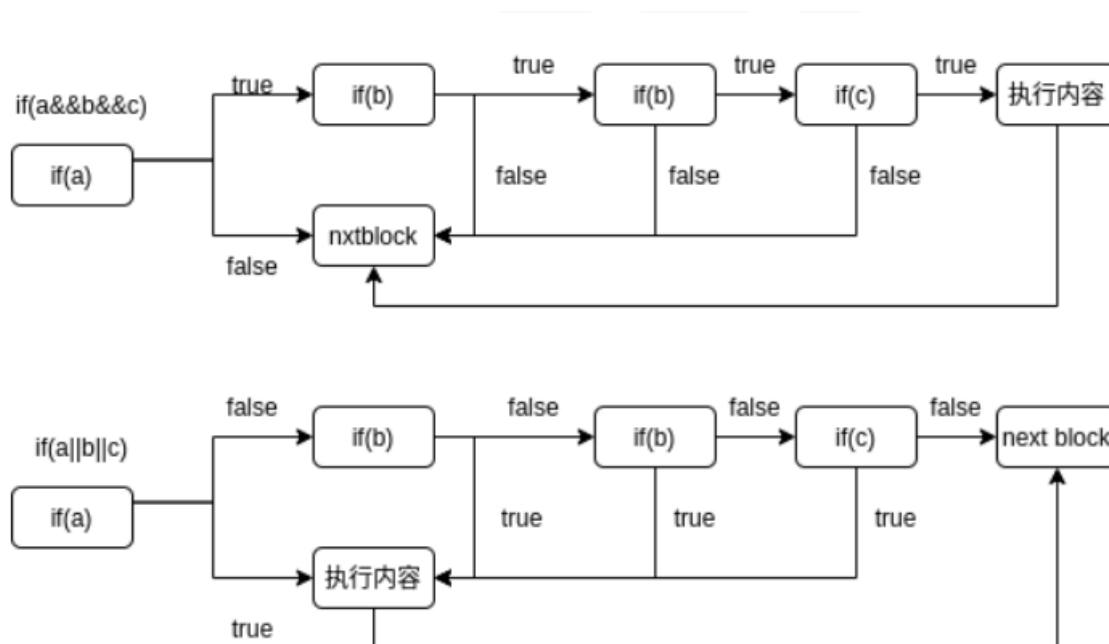
当出现and或or时，并不进行实际的逻辑与或运算，而是一个一个条件判断，当能确定值时马上跳转到对应代码块。

代码块的新跳转流程

当是逻辑与运算时，先判断前面的条件是否为true，是则继续判断，否则表达式一定是false，跳转到nextblock。

当是逻辑或运算时，先判断前面的条件是否为true，是则表达式一定是true，跳转到if内的执行语句，否则继续判断。

流程图如下：



具体细节

如何知道if内语句位置和下一块位置？

和Break、continue类似，采用**回填法**，先用标识符占位，当知道了对应位置再替换。

跳转的代码实现

1)在每一个&&、||出现后不进行运算，而是输出一条br语句，然后开辟一个新的代码块。

or的语句如下：

其中#代表if内部语句的位置，需要等待回填

```
codeblock.append("br i1 "+x1+", label %x#"+", label %x"+
(block_idx+1)+" "+x1+"\n");
block_idx++;
codeBlocks.add(new CodeBlock("x"+block_idx,new StringBuffer()));
```

and的语句如下:

其中`代表else后语句的位置, 没有else就是if语句结束后的位置, 需要等待回填。

```
codeblock.append("br i1 "+x1+", label %x"+(block_idx+1)+", label %x`\\n");
block_idx++;
codeBlocks.add(new CodeBlock("x"+block_idx,new StringBuffer()));
```

何时and如何回填?

由于或运算的优先级低于与运算, 因此与运算要在遇到或运算时立即回填, 同时在条件语句结束时也要回填。

而或运算只需要在整个条件语句结束后回填即可。

回填代码示例:

```
int tmp_l=block_idx;
LAndExp();
int tmp_r=block_idx;
for(int k=tmp_l;k<=tmp_r;k++){
    String tmps = codeBlocks.get(k).getResult().toString();
    tmps = tmps.replace("`", ""+(block_idx)); //回填相应的代码块id。
    codeBlocks.get(k).setResult(new StringBuffer(tmps));
}
```