

Stage-3 实验报告

吴垒 2020010916

实验内容

几乎完全是按照问答墙修改的，其实没啥好说的。

Step7 部分主要需要修改的是 `frontend/typecheck/namer.py`、`backend/dataflow/cfg.py` 及 `backend/reg/bruteregalloc.py`，对应符号表构建及寄存器的分配。

`Namer.py` 在 Step7 中需要修改：`visitBlock`，在访问一个块语句的时候为其创建一个局部作用域（`Scope`），在退出时还要将其关闭。

`Cfg.py` 在 Step7 中需要修改：添加 `unreachable` 函数，判断一个基本块是否可以抵达。具体判断方式：在控制流图初始化创建后使用 `dfs` 遍历从 0 号节点可以达到的所有节点，将其状态修改为可达即可。

`Bruteregalloc.py` 在 Step7 中需要修改：在 `accept` 函数中调用 `cfg.py` 中 `unreachable` 函数，判断当一个块不可抵达的时候直接跳过该块。

Step8 部分主要需要修改的是 `frontend/ast/tree.py`、`frontend/ast/visitor.py`、`frontend/lexer/lex.py`、`frontend/parser/ply_parser.py`、`frontend/typecheck/namer.py`、`frontend/tacgen/tacgen.py`。

`Tree.py` 在 Step8 中需要修改：添加 `For` 节点，含有 `init`、`cond`、`update`、`body` 子节点；添加 `DoWhile` 节点，含有 `body`、`cond` 子节点；`Continue` 节点，不含子节点。

`Visit.py` 在 Step8 中需要修改：添加 `visitFor`、`visitDoWhile`、`visitContinue` 函数。

`Lex.py` 在 Step8 中需要修改：添加 `for`、`do`、`continue` 的保留字。

`Ply_parser.py` 在 Step8 中需要修改：添加 `p_for`、`p_dowhile`、`p_continue` 函数（及其匹配产生式）。需要注意 `p_for` 的小括号内语句可以为空、也可以包含声明，因此应采用两条生成式，一条为 `opt_expression`，一条为 `declaration`，与 `statement` 的状态进行组合共四条。

`Namer.py` 在 Step8 中需要修改：添加 `visitFor`、`visitDoWhile`、`visitContinue` 函数，完全按照注释的步骤访问即可，仅需在进入循环时在栈空间 `openLoop`，退出时关闭即可。

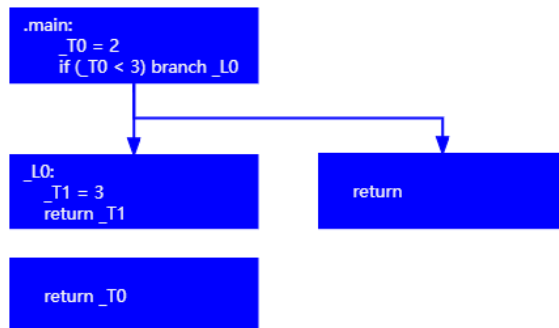
`Tacgen.py` 在 Step8 中需要修改：添加 `visitContinue`、`visitFor`、`visitDoWhile` 函数。需要注意的是在访问每个子节点前需先调用其 `accept` 函数计算 `val`，否则会产生错误。`For`、`dowhile` 可模仿 `while` 写，其中需注意 `for` 的括号内为空的情况；`Continue` 则与 `break` 相似。

一个检查时的偶然的小发现（源自本应是代码 bug 的东西）：`namer` 中的 `visitFor` 没有判断 `init`、`cond`、`update` 是否为空就可以直接调用其 `accept` 函数（即便为空也可以正确运行），进一步推断 `visitDeclaration` 也不需要判断 `init_expr` 是否存在即可直接调用其 `accept`，实验发现成立。通过 `raiseError` 将 `for` 内空时的 `init` 打印出来，发现此时 `init` 是 `NULL`（`node.py` 下的 `nulltype` 类）。

在反复阅读 `tree`、`parser` 程序后进行核理推断：在 `Declaration` 阶段 `init_expr` 默认为 `None`，但当语法分析 `p_declaration` 初始化 `Declaration` 仅有两个参数时由于 `Declaration` 的 `init` 函数有 `self.init_expr = init_expr or NULL`，而 `NULL` 其实是有实际值的，故此时 `init_expr` 被赋值成 `NULL`；在 `For` 阶段由于我在语法分析时采用的是 `opt_expression` 匹配，因此既可以在 `expression` 非空时调用 `p_opt_expression` 表达式产生 `expression`，又可以在其为空时调用 `p_opt_expression` 产生 `NULL`，因此 `init`、`cond`、`update` 在不存在时也会被赋值为 `NULL` 而不是 `None`。而 `NULL` 是含有 `accept` 属性的，因此可以直接调用 `accept` 函数而不报错。

思考题

Step7: 控制流图



Step8: while 的翻译方式

第二种方式更好。对于相同的 `while` 程序，第二种翻译方式每一轮会比第一种少一条指令，因为当条件仍满足循环要求时，第二种的 `bnez` 同时完成了比较与跳转的任务，而第一种则需要用 `beqz` 完成比较任务，`branch` 完成跳转任务。

借鉴内容

实验思路指导与问答墙