

hw6实验过程记录

参考资料

代码组织参考 `hw5/docs/lab5.md`。

代码样式和流程参考 `hw7/include/frontend/semant.h`。

继承环的检测算法参考[Detect Cycle in a directed graph using colors](#)。

实验过程

代码组织：所有的函数声明放在 `semant.h`，实现放在 `semant.c`。

utils

- 在 `semant.h/c` 维护表达式类型 `expty`：除了 `Ty_ty`，还要维护是否为左值 (即 `bool location;`)

```
1 typedef struct expty_* expty;  
2 struct expty_ {  
3     bool location;  
4     Ty_ty ty;  
5 };
```

- 在 `semant.h/c` 声明一系列辅助函数：

```
1 /* 辅助函数 */  
2  
3 // 用于检查父类和子类的同名函数的签名是否严格相同：返回值类型 + 参数类型  
4 bool equalClassMethSignature(E_enventry fa, E_enventry cur);  
5  
6 // 检查 Ty_ty 严格相等  
7 // 用于签名检查  
8 bool equalTy(Ty_ty fa, Ty_ty cur);  
9  
10 // 检查 Ty_ty 相等：允许 int<=>float 和 upcast  
11 // 用于函数调用和返回语句的类型检查  
12 bool equalTyCast(Ty_ty param, Ty_ty arg);  
13  
14 // 要求 right 是 left 的子类或本身  
15 bool isParentClass(Ty_ty left, Ty_ty right);  
16  
17 // AST 类型 --> semant 类型  
18 Ty_ty atype2tyty(A_type t);  
19 Ty_field f2tyf(A_formal f);  
20  
21 // 便于错误报告  
22 string ty2str(Ty_ty t);
```

```

23
24 // 由于VarEntry需要vd，故在检查方法时需将A_formal->A_varDecl
25 // 用于将方法参数插入变量表中
26 A_varDecl f2vd(A_formal f);
27
28 // 用于格式化字符串的声明
29 // 便于输出更详细的错误信息
30 string sprintf(char *, ...);

```

代码流程

类型检查主要可分为以下三个阶段：

1. 初始阶段：声明需要用到的静态变量
2. 预处理阶段：对所有的类声明遍历2次，其中
 - 第一次遍历：主要是建立cenv表
 - 第二次遍历：主要是检测继承环，并将父类的所有类变量和类方法复制到子类的相应表中
3. 类型检查阶段：检查类和主方法
 1. 遍历类：检查类变量和类方法
 2. 检查主方法

初始阶段

本次实验要用到的静态全局变量如下：

1. **cenv**：类环境，存储类名到类信息的映射，其中类信息中包括2个重要的表
 - **vtbl**：类变量表，存储类变量声明
 - **mtbl**：类方法表，存储类方法声明
2. **venv**：变量环境，在对类方法和main method进行类型检查时存储声明的变量 (和参数)
3. **MAIN_CLASS**：一个dummy class的名字，主要作用有2个：
 - 在初始化cenv时，用作所有无extend的类的父类
 - 在检查主方法时，用作curClassId，从而区分主方法和其他方法
4. **curClassId**：目前正在检查的类名，在预处理阶段和类型检查阶段使用
5. **curMethodId**：目前正在检查的方法名，在预处理阶段和类型检查阶段使用

预处理阶段

该阶段需要对所有类遍历2次：

1. 初始化cenv
 1. fa：记录父类，即extends关系

2. vtbl: 将所有变量插入表中, 同时检查是否有重定义

3. mtbl: 将所有方法插入表中, 同时检查是否有重定义

- 记录返回值类型
- 记录参数类型: 在遍历参数列表, 返回参数类型的同时检查参数是否重定义
 - 将参数按A_varDecl的形式插入venv中
 - 伪插入: 用S_beginScope(venv)和S_endScope(venv)包裹, 检查完毕后可以删除所有插入的参数

2. 继承环检测

○ 检测算法: 将类继承关系看作有向图 ($A \text{ extends } B \Leftrightarrow A \rightarrow B$), 则问题转化为检测有向图中是否有环

- 每个节点 (类) 都有以下3种颜色中的一种:
 1. white: 节点还未处理, 即一开始所有节点都是白色
 2. gray: 正在处理该节点, 但还未处理完它的所有父类
 3. black: 该节点及其所有父类都已经处理完
- DFS算法:
 1. 将当前节点标为grey
 2. 若当前节点的父类是grey, 说明出现继承环
 - 同时这里可以检测父类未定义的错误
 3. 若当前节点的父类是white, 递归地处理当前节点的父类
 4. 待所有父类都处理完后, 将当前节点标为black
 - 同时这里可以将父类的vtbl和mtbl复制到当前节点, 如此回溯就可以将所有父类的变量和方法复制到子类
 - 在复制的过程中, 可检查类变量是否重名, 报错重定义
 - 在复制的过程中, 可检查同名类方法的函数签名 (返回值类型+参数类型) 是否严格相同, 若不同则报错
 - 无需将父类的同名方法复制到子类的mtbl中

复制的遍历过程: 利用binder->prevtop反向遍历, 利用getBinder API获取值

- 总算法: 对每条路径依次检测, 即若当前类是white就运行DFS算法进行环检测
- 这里的3种颜色可以与E_status的状态一一对应:
 - white \leftrightarrow E_transInit
 - grey \leftrightarrow E_transFind
 - black \leftrightarrow E_transFill

类型检查阶段

该阶段的主要流程如下：

1. 遍历类

1. 检查类变量：只需要对类型为class的变量，查询cenv看这个class是否存在
2. 检查类方法：进入时S_beginScope(venv)，检查退出后S_endScope(venv)
 1. 检查返回值类型：若类型为class，查询cenv看这个class是否存在
 2. 再次检查参数类型：
 - 只需要对类型为class的参数，查询cenv看这个class是否存在
 - 同时，将参数按A_varDecl存入venv中
 3. 检查变量声明：和参数检查类似，同时要求和其他变量、参数都不能同名
 4. 检查Stms

2. 检查主方法：完全类似类方法的检查，但此时curClassId设为MAIN_CLASS

对Stms和Exps的类型检查，要注意以下几个方面：

1. 类型匹配：在赋值语句、函数调用和返回语句中，需要检查类型是否匹配
 - 允许int和float之间的隐式转换
 - 不允许int[]和float[]之间的转换
 - 允许将子类对象赋给父类对象：利用工具函数isParentClass(leftTy, rightTy)检查
2. CallStm和CallExp：需要检查形参和实参列表的长度和类型是否匹配
 - 同时遍历filedList和expList，若长度不匹配 (即只有其中一个为NULL)，报错pos应该是对应的Stm/Exp的pos
 - 规定调用transA_ExpList_Call()时，必须传入对应的Stm/Exp的pos
 - 若某个形参和实参类型不匹配，报错pos是该实参的pos

测试结果

测试结果如下：

```
● zqwh@LAPTOP-HDCBVNK7:~/compiler/2024/hw6$ make test
[hw6test00]
(line:3 col:1) error: class C2 has a cycle in inheritance
[hw6test01]
(line:7 col:29) error: class c5 has duplicate variable names with parent class
[hw6test02]
[hw6test03]
(line:8 col:12) error: class c0 has no method m
[hw6test04]
(line:8 col:5) error: return value of main method must be of type int or float
[hw6test05]
(line:31 col:5) Error: Class c1 has no variable x4
[hw6test06]
(line:21 col:5) error: object types do not match in assignment, right side expected 'class c1', got 'class c4'
[hw6test07]
(line:10 col:9) error: return type expected 'class c2', got 'class c1'
[hw6test08]
(line:3 col:5) error: object types do not match in assignment, right side expected 'class c2', got 'class c1'
[hw6test09]
[hw6test10]
(line:19 col:21) error: method call argument types do not match, expected 'int', got 'array of int'
[hw6test11]
(line:12 col:22) error: variable not declared
[hw6test12]
(line:19 col:9) error: variable not declared
[try]
(line:3 col:27) error: break statement outside of loop
```

```
● zqwh@LAPTOP-HDCBVNK7:~/compiler/2024/hw6$ make astcheck
[hw6test00]
(line:3 col:1) cyclic class extend!
[hw6test01]
(line:7 col:29) Redclared class variable!
[hw6test02]
[hw6test03]
(line:8 col:12) Undefined class method!
[hw6test04]
(line:8 col:5) Main method should return a num!
[hw6test05]
(line:31 col:5) Undefined class variable!
[hw6test06]
(line:21 col:5) Uncompatible class type!
[hw6test07]
(line:10 col:9) Uncompatible class type!
[hw6test08]
(line:3 col:5) Uncompatible class type!
[hw6test09]
[hw6test10]
(line:19 col:21) Uncompatible type!
[hw6test11]
(line:12 col:22) Undefined variable!
[hw6test12]
(line:19 col:9) Undefined variable!
[try]
(line:3 col:27) Break must be inside while!
```

开发过程

开发过程基本按实验过程进行：(以下提交记录只是一部分)

hw6: refactor	10 Apr 2024 16:54	Jopqior	73977d51
hw6: change Makefile	10 Apr 2024 11:33	Jopqior	1d335078
hw6: remove duplicate tools	10 Apr 2024 11:13	Jopqior	ff2732ad
hw6: change Makefile	10 Apr 2024 10:36	Jopqior	6cfb6a88
hw6: fix while depth and change some output style	10 Apr 2024 09:52	Jopqior	acb6b6c6
hw6: fix formal check and change Makefile	10 Apr 2024 09:47	Jopqior	6020a7aa
hw6: fix output pos in cycle detect and change method fl reded to phase1	9 Apr 2024 23:18	Jopqior	2288e685
hw6: merge master	8 Apr 2024 21:23	Jopqior	e682a20f
hw6: remove hw1 hw3_4 hw5	8 Apr 2024 21:16	Jopqior	61763bc5
hw6: add external test	7 Apr 2024 21:58	Jopqior	320fb1aa
hw6: update astcheck	7 Apr 2024 20:32	Jopqior	5ae838f1
hw6: fix bug that must this.*	7 Apr 2024 20:03	Jopqior	dcadd62b
hw6: fix while loop counter	7 Apr 2024 17:21	Jopqior	0f38f645
hw6: update astcheck	7 Apr 2024 17:21	Jopqior	5e807cc2
hw6: change output style	7 Apr 2024 17:05	Jopqior	7277185c
hw6: merge hw5	7 Apr 2024 11:21	Jopqior	a539ff0a
hw6: chmod	6 Apr 2024 17:40	Jopqior	46d9192a
hw6: merge master	6 Apr 2024 16:36	Jopqior	225e4b13
hw6: merge hw5	6 Apr 2024 16:33	Jopqior	3d225ebf
hw6: change output style	6 Apr 2024 14:28	Jopqior	77dc432a
hw6: change output for method call type checking	6 Apr 2024 14:11	Jopqior	220fed1b
hw6: merge master	6 Apr 2024 11:53	Jopqior	db064f26
hw6: copy class vtbl to venv everytime type checking in a class	6 Apr 2024 11:51	Jopqior	eec5d90d
hw6: complete exp	5 Apr 2024 14:04	Jopqior	191c0cf7
hw6: add ty2str and complete return stm	5 Apr 2024 13:07	Jopqior	f2b94422
hw6: fix exp	5 Apr 2024 12:45	Jopqior	4219d05d