编译原理专题训练大作业

Lustre 及其 AST 双向翻译器

项目报告

2012011894 朱俸民

2011011779 孙皓

2012011487 徐梓哲

一、项目背景

Lustre 是一门工业上使用的语言，适宜编写针对数据流的程序，由于可靠程度比较高，经常被用于高安全级别的场合。

Lustre 的语法本身功能较弱，为了方便查看，实验室打算将其翻译成 C 语言。为了实现这个目标，需要先构造中间文件 AST，即抽象语法树。

二、项目内容

我们的工作一开始是实现 AST 到 Lustre 的翻译器，之后随着项目的进展，发现以前的 Lustre 到 AST 的翻译器根本无法生成合法的 AST 文件，因此又实现了 Lustre 到 AST 的翻译器。

两个方向的翻译器有着大致类似的流程。项目由 OCaml 编写，翻译过程首先使用 OCaml 的 lexer 和 yacc 工具构造树结构，然后根据树结构构造目标语言文件。

由于语法上的差别，Lustre 到 AST 的过程，在输出目标文件的时候还构造了符号表，完成了预计算与符号的预推演。

另外，我们还修改了 Lustre 与 AST 的语法文档，目前的文档在正确性、完整性方面与测例完全契合。

项目保存在 github，地址为 https://github.com/paulzfm/LustreAST；

课程网页中上传的文件即为 github 项目的所有内容。

三、项目结果

基于测例，我们对两个翻译器进行了测试，结果如下：

1、AST 到 Lustre 翻译器

由于 OCaml lex 工具的符号表规模有限，我们使用 python 脚本对输入 lustre 文件进行了预处理，以完成词法分析，因此程序的使用方式为：

```
python lexer.py -i <ast_file> | ./ast
```

需要注意的是，由于需要用到辅助文档，这条命令需要在程序所在目录下运行，请注意运行路径。

考虑到 lustre 中括号可以随意增加，且空白符并没有什么意义，我们去除了所有的’(‘,’)’,’\n’,’\r’,’ ‘,’\t’。

另外，Lustre 中函数开头有 function 与 node 两个关键字表示函数的类型，但 AST 测例并没有保持一致，一个系列测例与 Lustre 文件相同，一个系列的测例 AST 文件全都是 function。除此之外，Lustre 中有 private 关键字，但 AST 中没有这个语法考虑。

为了减少这方面的比较负担，我们去除了所有的 private，function 和 node 关键字。

比较结果如下：

209 个测例完全匹配；其中 private 差别与括号差别在测试结果文件里有所说明；

17 个测例以及特别长的 firmsyslib_assembly.ast 顺利产生 Lustre 结果，但缺少 Lustre 测例；

24 个测例不完全匹配，原因如下：

23 个测例存在常数差别。包括两种情况，一种是由精度产生，比如 3.4 在 AST 中有时会被保存为比 3.4 差一个非常小的数，这是由浮点数精度引起的；另一种是预计算的问题，比如 Lustre 中可能是 1.5+2.1，在 AST 中会被预计算，直接记录结果 3.6；很明显这两种都不可能解决；

1 个测例错误，srs_l2c_syn_049_025 的 Lustre 文件与 AST 文件明显不是同一个程序。

具体测试结果列表请查看提交的测试结果统计，ast2lustre/tests/test_report.txt；

2、Lustre 到 AST 翻译器

Lustre 语言的关键字比较少，这个方向的翻译器完全由 OCaml 编写。使用方式为：

```
./lustre < <lustre_file>
```

明显不受运行路径影响。

与上述类似的，我们比较 AST 测例以及我们的输出文件时，删除了’\n’,’\r’,’ ‘,’\t’,’function’,’node’，由于 AST 结构性较强，括号可以纳入比较的考虑范围。

比较结果如下：

219 个测例完全匹配；

11 个测例缺乏 AST 测例文件无法比对；

13 个测例不完全匹配，原因如下：

8 个测例存在常数不匹配问题。这次只有浮点数精度问题，预计算由于 Lustre 到 AST 是信息损失，这个翻译器可以完成；

3 个测例存在主函数不匹配问题。Lustre 文件只是单纯的堆叠函数，但 AST 文件会认为存在一个主函数，而且这个主函数的在 Lustre 文件中的位置并不固定，因此很明显没法保证一直匹配；

1 个测例存在类型未展开问题。AST 中定义的类型，在被使用时，会以其展开形式，比如基本类型的数组的形式，被写出来，而不是定义的那个类型的名字。由于 AST 实际上支持使用文件内定义的类型，这里的差异只是文本上的差异，语义上实际等价。

1 个测例同时出现主函数不匹配与类型未展开问题。

值得说的是，我们的翻译器生成了符号表，进行了预计算与类型预推演，实际上在大多数测例下是会对类型进行适当的展开的，而且在所有测例中都完成了常数的预计算，因此出现的问题只是少数，原因不明。

具体测试结果列表请查看提交的测试结果统计，lustre2ast/tests/test_report.txt；

四、项目分工

AST 到 Lustre 翻译器：

朱俸民：目标语言输出代码，语法研究，测试；

孙皓：语法分析代码，语法研究与文档修正，测试；

徐梓哲：词法分析代码；

Lustre 到 AST 翻译器：

朱俸民：符号表推演及预计算代码，目标语言输出代码，测试；

孙皓：语法研究与文档修正，词法分析与语法分析代码，测试；

徐梓哲：注释处理