

COL 226: Assignment 3

1 Introduction

This assignment requires you to define abstract syntax for the simple language PL0, build a parser component to translate parse trees of PL0 programs into Abstract Syntax Trees (ASTs) and a pretty printer that can print out ASTs of parsed programs in a nice format.

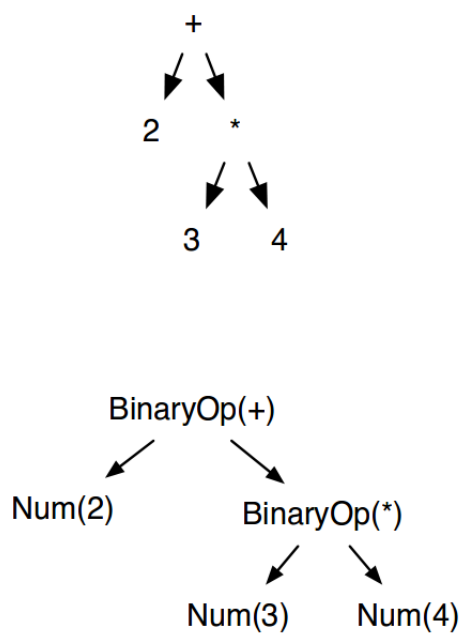
2 Parsing

Parsing involves converting an input sequence of text, such as a program, into a tree structure according to a grammar. The grammar for a language describes the concrete syntax of programs, including all the tokens that are part of the input. Rather than to construct the concrete syntax tree, the job of the parser is to build an abstract syntax tree.

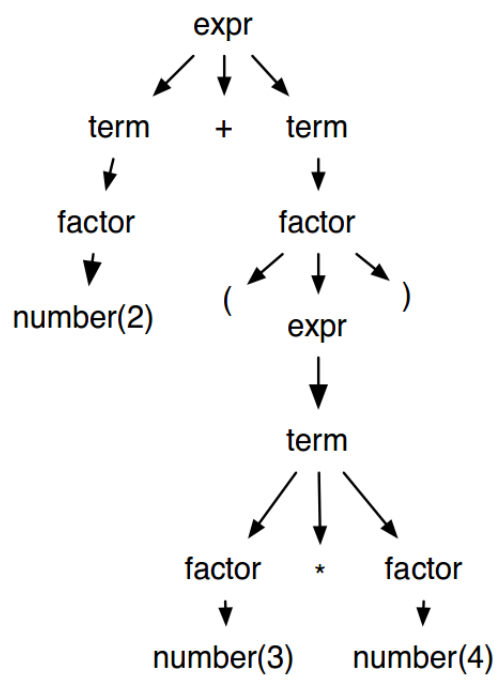
You will apply this same idea to parsing a program written in a PL0 language into an internal abstract syntax tree representation that your program can understand, execute, and modify. So, in this assignment your task is to translate the parse trees generated by your second assignment into ASTs.

2.1 Abstract syntax trees

An abstract syntax tree (AST) represents the given input without parts that do not affect its meaning, inherent in the structure of the AST. As a result, different inputs that have the same meaning will have an identical AST. For example, the expressions $(2 + 3 * 4)$, $2 + (3 * 4)$, and $(2) + (3) * (4)$ have the same abstract syntax tree, because parentheses are only there to guide the construction of the tree. Figure 1 shows this abstract syntax tree along with the concrete syntax tree (parse tree) for $2 + (3 * 4)$. The AST is shown on the left in two different forms: the top represents how we might think of the AST, while the bottom corresponds more closely to the code and might help with your AST implementation. The abstract syntax tree omits any syntax that is unnecessary, which makes it different from a concrete syntax tree (parse tree). For this, you will need to define suitable data types in SML/OCaml.



abstract syntax tree



parse tree

Figure 1: Abstract and concrete syntax trees for $2+(3*4)$

3 Instructions for Submission

- All submissions must be through moodle. No other form of submission will be entertained.
- No submissions will be entertained after the submission portal closes.
- Sometimes there are two deadlines possible – the early submission deadline (which we may call the "lifeline") and the final "deadline". All submissions between the "lifeline" and the "deadline" will suffer a penalty to be determined appropriately.

4 What to Submit?

- You will create one folder which will have 2 files, program file and the writeup file.
- The program file should be named with your Kerberos ID. For example, if kerberos id is 'cs1140999' then the file name should be cs1140999.sml or cs1140999.ocaml. The writeup should be named as "writeup.txt".
- Both the files should be present in one folder. Your folder also should be named as your Kerberos ID. For example, if kerberos id is 'cs1140999' then the folder should be called cs1140999.
- The first line of writeup should contain a numeral indicating language preferred with 0-ocaml, 1-sml and 2-haskell.
- For submission, the folder containing the files should be zipped(".zip" format). Note that, you have to zip folder and NOT the files.
- This zip file also should have name as your Kerberos ID. For example, if kerberos id is 'cs1140999' then the zip file should be called cs1140999.zip.
- Since the folder has to be zipped the file cs1140999.zip should actually produce a new folder cs1140999 with files (cs1140999.sml or cs1140999.ocaml) and writeup.txt.

Hence the command *"unzip -l cs1140999.zip"* should show

```
cs1140999/cs1140999.sml
cs1140999/writeup.txt
```

- After creating zip, you have to convert ".zip" to base64(.b64) format as follows (for example in ubuntu):
base64 cs1140999.zip > cs1140999.zip.b64 will convert .zip to .zip.b64
This cs1140999.zip.b64 needs to be uploaded on moodle.
- After uploading, please check your submission is up-to the mark or not, by clicking on evaluate. It will show result of evaluation. If folder is as required, there will be no error, else REJECTED with reason will be shown. So, make sure that submission is not rejected.