CMPT 308 - Spring 2024 - Dr. Labouseur

Lab 6/7 – Lexer

Patrick Tyler
Patrick.Tyler1@marist.edu

April 8, 2024

1 Crafting A Compiler (lab 6)

1.1 8.1 BST and Hash Tables

Hash tables have more upfront cost than binary search trees both in terms of computation and space (depending on the capcity). This is due to the hashing process and need to store spots for each possible hash results. In small symbol name spaces binary search is probably better as the few extra comparisons is faster than the implemented hasing algorithm. For adding symbols in binary search it can be done in O(log n) time assuming space in the array is left open else O(n). Getting is also O(n). Hash tables can add and remove in O(n + load factor) but as said before have more up cost depeding on capacity and hashing algorithm

2 Explain what is happening (lab 7)

The diagram below shows an AST and indicates that an abtract production (print parent of b) is being processed. This production in the AST maps to the code highlighted in the source code ”print(b)”. In the process of ensuring that b is valid in the namespace, it is search for in the scope tree of hashamps. This print is happening in scope 1b so the first scope checked is 1b. The symbol ”b” is not in scope 1b so its parent scope 0 is then checked. Symbol ”b” is found in scope 0 so it is marked that the the b from that print references the b declared in scope 0. It is important to note that only scope 1b and its parents would be searched for the earliest declaration of symbol ”b”. The siblings of 1b like 1 would not be searched as those scopes are not included in 1b as that is how block scope works.