

CS 381 - A7

Martin Mueller

Due: April 17th 2020

We start by constructing a grammar for the language of P :

$$\begin{aligned} S &\rightarrow A \mid B \mid C \\ A &\rightarrow \{AB\} \mid AA \mid \lambda \\ B &\rightarrow [BC] \mid BB \mid \lambda \\ C &\rightarrow (C) \mid CC \mid \lambda \end{aligned}$$

This grammar enforces precedence by having different levels of symbols arranged such that it is impossible to go up a level, but it is possible to go down levels so that the correct brackets are contained within each other. This grammar also allows for starting at any level and accepts the empty string.

Now that we have a grammar, we can construct a PDA that recognizes it. First, we initialize the stack by pushing a dollar sign onto it. Next, we push the start symbol S onto it, then nondeterministically branch out until we reach the correct combination of terminals on the stack. Once the input matches the stack, then we pop all the terminals on the stack until we hit the dollar sign. When that happens, we transition into an accept state. Notice that the PDA's stack shares the same alphabet as the grammar and the input alphabet is exactly the set of terminals in the grammar.

