Software Specifications
Converting Grammar to Push Down Automaton

Cain Susko

Queen's University
School of Computing

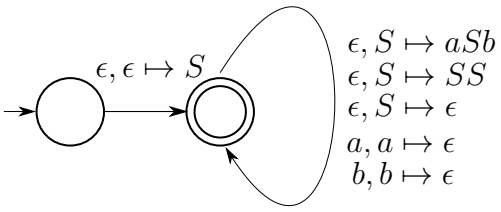
February 7, 2022

Grammar to Push Down Automaton

We shall use the following parenthesis grammar to demonstrate how to convert a grammar to a corresponding push down automaton.

$$S \to aSb \mid SS \mid \epsilon.$$

This grammar generates the set of all well nested parenthesis sequences; where a is the left parenthesis and b is the right parenthesis.
We will thus create a 2 state non-deterministic automaton: to show how this



PDA works, we will show a computation given the string *abab*. We must first show that *abab* can be derived from the grammar:

$$S \implies SS \implies aSbS \implies abS \implies abaSb \implies abab.$$

Thus the computation table for this Automaton given the string *abab* is:

state	stack	remaining input
1	ϵ	abab
2	S	abab
2	SS	abab
2	aSbS	abab
2	SbS	bab
2	bS	bab
2	S	ab
2	aSb	ab
2	Sb	b
2	b	b
2	ϵ	ϵ

Note that this machine is highly non-deterministic (the self loop on the final state) and thus requires alot of decisions to be made by the human depending on the desired result within this grammar.

It is also possible to convert a PDA to a grammar, however this is not covered in this course. It is also important to note that non-deterministic PDA's recognize the same class of languages that are generated by context free languages. Furthermore, some context free grammar languages do not have a corresponding deterministic PDA