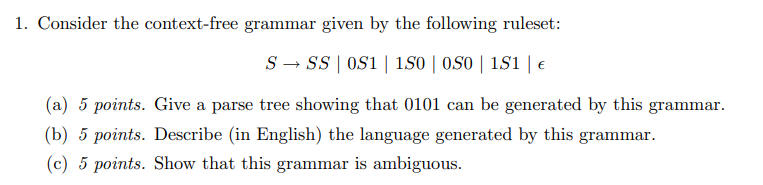
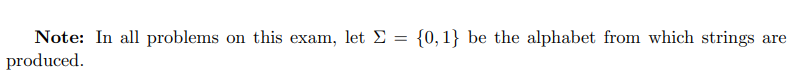
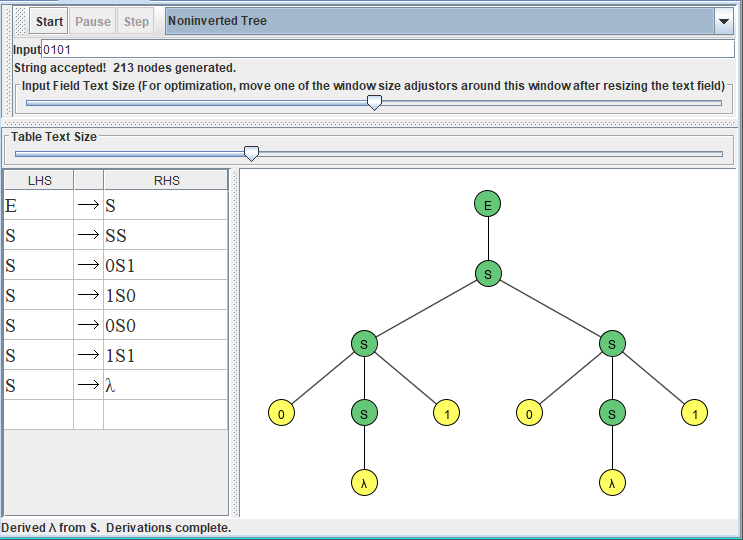
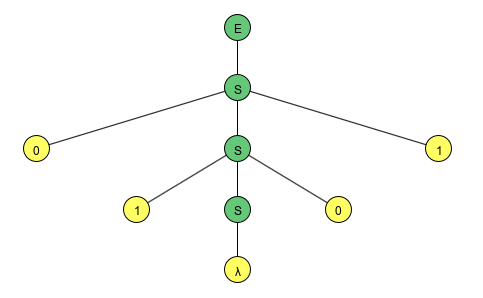
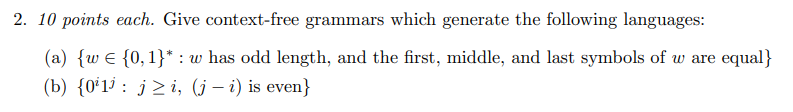
Colin Quinn

Prof. Huggins

CS 312 Exam 2



1. 
2. The language described here is any even length binary string, including the empty string. Can also be expressed as the expression (0 + 1)\*.
3. The string 0101 can also be expressed as this solution: 



1. E → S

S → 0 | 1 | 0A0 | 1B1

A → 0A0 | 0A1 | 1A0 | 1A1 | 0

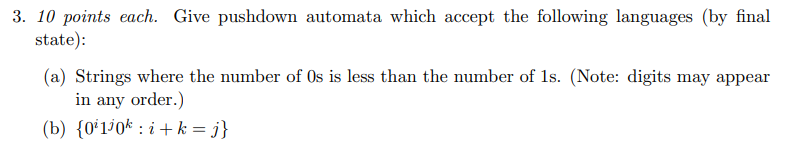
B → 0B0 | 0B1 | 1B0 | 1B1 | 1

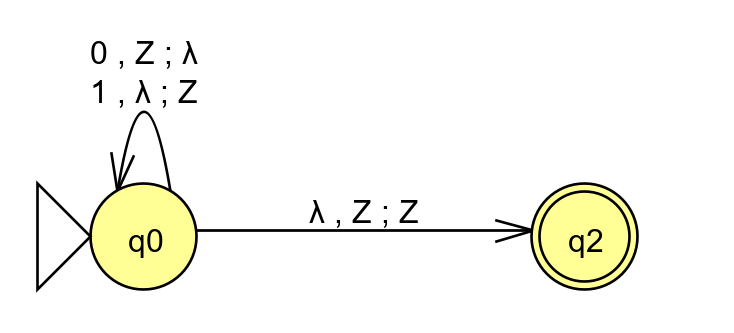
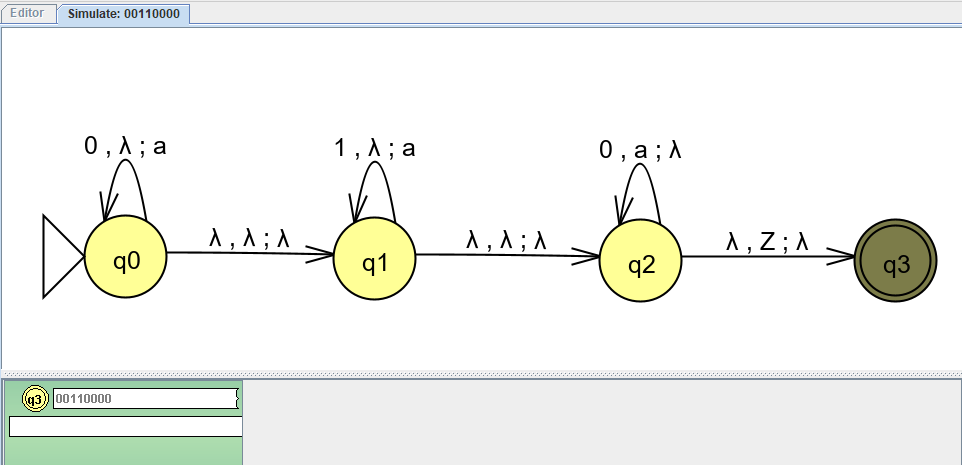
1. E → S

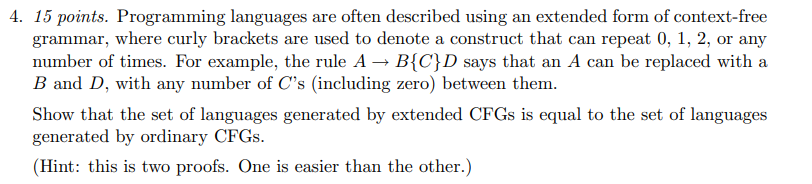
S → A | B

A → 0A11 | A1 | 1

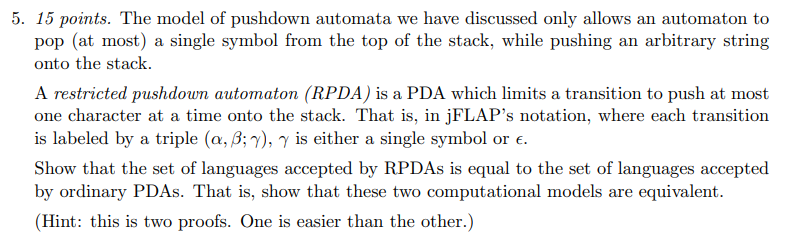
B → B11 | 0



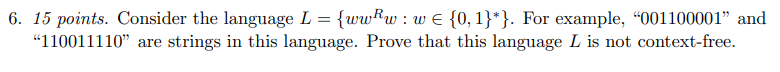
1. 
2. 



An extended context-free grammar is a CFG that can be expressed as a regular expression. This definition of A → B{C}D only allows for context-free grammars to be generated due to its definition as A → B(C\*)D where C can occur any number of times.



All RPDA’s are still by definition PDA’s. Additionally, all PDA’s can be converted into RPDA’s by adding additional transitions that pop additional characters that are not epsilon onto the stack. That is, have two separate transitions stating *1, epsilon; a* as well as *1,* epsilon*; s* in order to push the string *as* onto the stack.



Let’s assume that this language L is context-free. Via the pumping lemma, some *R* exists such that for any string x ∈ L while |x| >= *R*. Let u = w and y = wR, thus uyu ∈ L, applying the pumping lemma would result in the string uy2u !∈ L.