70/70 + (100/100)

CS 3350 Automata, Computability, and Formal Languages
Fall 2017, Test 2

Name: ______________________________

General comments:

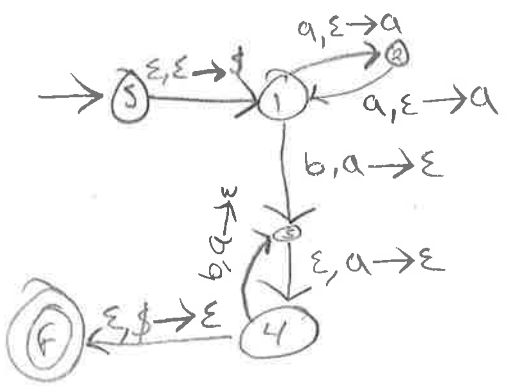
- you are allowed up to 5 pages of handwritten notes;
- if you need extra pages, place your name on each extra page;
- the main goal of most questions is to show that you know the corresponding algorithms; so, if you are running out of time, just follow the few first steps of the corresponding algorithm.

Good luck!

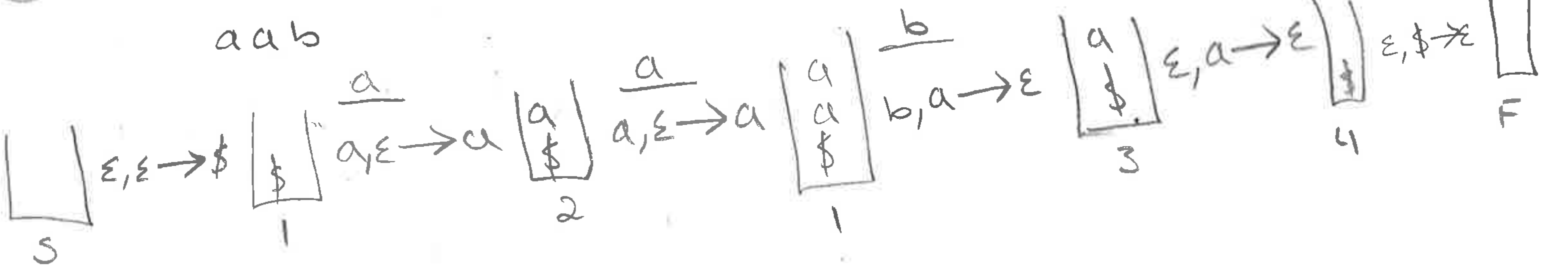
10/10

1. Design a pushdown automaton that would recognize the words of the type $a^{2n}b^n$, i.e., the language $L = \{\Lambda, aab, aaaabb, ...\}$. Show, step by step, how your pushdown automaton will recognize the word aab. *Hint*: use a pushdown automaton for recognizing the words of the type a^nb^n as a sample. The difference is that in that case, when we saw a letter 'b', we popped one symbol 'a' from the stack. Now it is slightly different.

$a^{2n}b^n$



aab

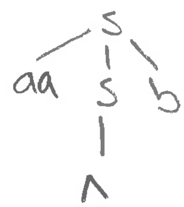


10|10

2. Design a context-free grammar that would generate all the words of the type $a^{2n}b^n$, i.e., the language L = {Λ, aab, aaaabb, ...}. Show, step by step, how your grammar will generate the word aab. *Hint*: use a context-free grammar for generating the words of the type $a^n b^n$ as a sample. There, we had rules $S \to \varepsilon$ and $S \to aSb$, now a slight modification is needed.

$a^{2n} b^n$ aab

$$\boxed{\begin{array}{l} S \to \Lambda \\ S \to aaSb \end{array}}$$

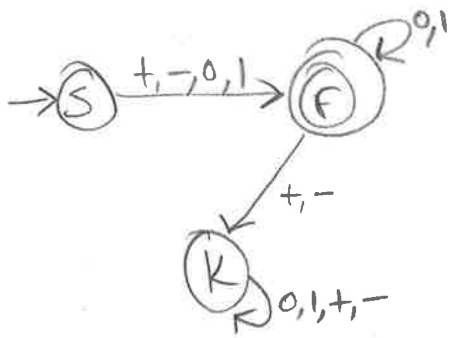


10/10

3. Use a general algorithm that we had in class to generate a context-free grammar that corresponds to the following finite automaton for recognizing signed or unsigned binary integers. This automaton has three states: the stating state s, the final state f, and the sink state k.

- From s, any symbol (+, −, 0, or 1) lead to f.
- From f, 0 or 1 lead to f, while + or − lead to sink.

Show, step by step, how your grammar generates a string +01.



$S \to +F$ $F \to 0F$ $K \to 0K$

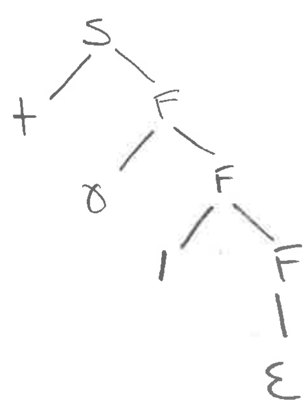
$S \to -F$ $F \to 1F$ $K \to 1K$

$S \to 0F$ $F \to +K$ $K \to +K$

$S \to 1F$ $F \to -K$ $K \to -K$

$F \to \varepsilon$

+01



10/10

4. Transform, step by step, the grammar with rules S → ε and S → 1S0 to Chomsky normal form. Show how the word 10 will be generated in the resulting Chomsky-normal-form grammar.

$S \to \varepsilon$

$S \to 1S0$

Pre

$S_0 \to S$

~~$S \to \varepsilon$~~

$S \to 1S0$

step 0:

~~$S_0 \to S$~~

$S_0 \to \varepsilon$

$S \to 1S0$

$S \to 10$

Step 1:

$S_0 \to 1S0$

~~$S_0 \to 10$~~

$S_0 \to \varepsilon$

$S \to 1S0$

~~$S \to 10$~~

step 2:

$V_0 \to 0$

$V_1 \to 1$

~~$S_0 \to 1S0$~~

$S_0 \to V_1 V_0$

$S_0 \to \varepsilon$

~~$S \to 1S0$~~

$S \to V_1 V_0$

step 3:

$V_0 \to 0$

$V_1 \to 1$

$S_0 \to V_{1S} V_0$

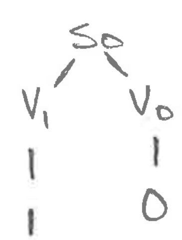
$V_{1S} \to V_1 S$

$S_0 \to V_1 V_0$

$S_0 \to \varepsilon$

$S \to V_{1S} V_0$

$S \to V_1 V_0$

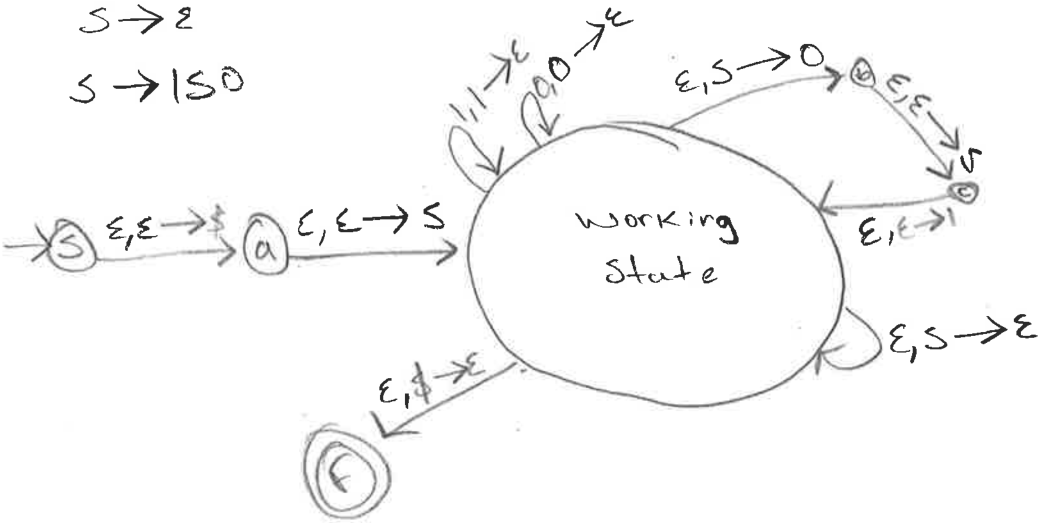


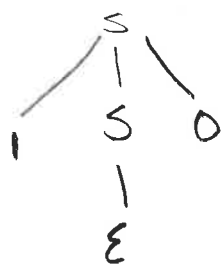
10/10

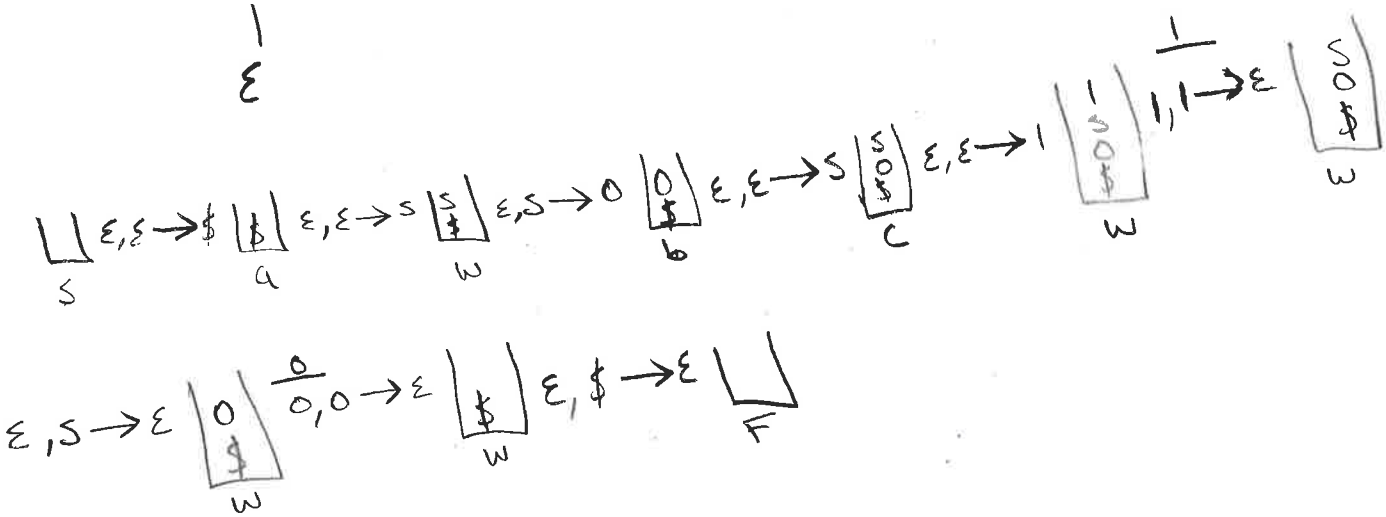
5. Use a general algorithm for transforming CFG into PDA to design a pushdown automaton which is equivalent to the the grammar with rules S → ε and S → 1S0. Show, step by step, how the word 10 will be accepted by the resulting pushdown automaton.

$S \to \varepsilon$

$S \to 1S0$



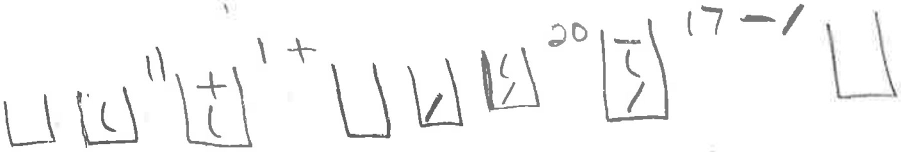




10/10

6. Use the general stack-based algorithms to show:

- how the compiler will transform the expression (11 + 1) / (20 − 17) into postfix form, and
- how it will compute the value of the resulting postfix expression.



$\boxed{11\ 1\ +\ 20\ 17\ -\ /}$

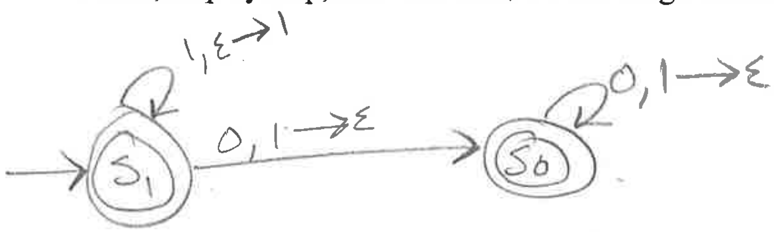
11 1 + 20 17 − /

10/10

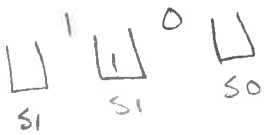
7. Use a general algorithm for transforming PDA into CFG to design a CFG that corresponds to the following pushdown automaton. This automaton has two states: the 1-state s_1 and the 0-state s_0. Both states are final. The transitions are:

- From s_1 to s_1, the transition is $1, \varepsilon \to 1$.
- From s_1 to s_0, the transition is $0, 1 \to \varepsilon$.
- From s_0 to s_0, the transition is $0, 1 \to \varepsilon$.

Show, step by step, how the word 10 will be generated by the resulting grammar.



10



$A_{pp} \to \varepsilon$

$A_{pq} \to A_{pr} A_{rq}$

$A_{s_1 s_0} \to 1 A_{s_1 s_1} 0$

$A_{s_1 s_0} \to 1 A_{s_1 s_0} 0$

$A_{s_1 s_1} \to \varepsilon$