CS3350 – Automata **Exam 2**
November 24, 2014 – 1 hour and 20 minutes

NAME: SOLUTION

GRADE: Total = / 100

1.	**/ 30**	**(/ 5; / 8; / 7; / 10)**
2.	**/ 30**	**(/ 10; / 10; / 10)**
3.	**/ 30**	**(/ 10; / 10; / 5; / 5)**
4.	**/ 10**	**(/ 5; / 5)**

Rules: This exam is to be done individually. Class notes and books of any kind are not allowed.
You must write legibly and in a structured manner: unreadable answers will be graded 0; messy / hard-to-follow answers will be penalized by up to 75%.

Cell phones should be turned off, no headsets allowed. Turn in your exam by 10:20am sharp: no exam will be accepted past this time.

---

1 General Knowledge [30 points]

1.1 What is a language? [5 points] – Answer in one sentence max.

A language is a set (of strings / sequences of characters or general inputs).

1.2 Consider the language L described by its regular expression: $(ab^+Ua^*b).ab$, where $\sum = \{a, b\}$. For each of the following strings, answer whether or not they are members of L. [8 points]

abbab	(YES)	NO
abb	YES	(NO)
ab	YES	(NO)
aaaab	YES	(NO)

1.3 Please circle the appropriate answer. [7 points]

- **There are Regular Languages that are not Context-Free languages:** TRUE (FALSE)
- **There exists a CF Grammar for every Regular Language:** (TRUE) FALSE
- **PDAs recognize regular languages:** (TRUE) FALSE
- **Turing Machines recognize Context-Free languages:** (TRUE) FALSE

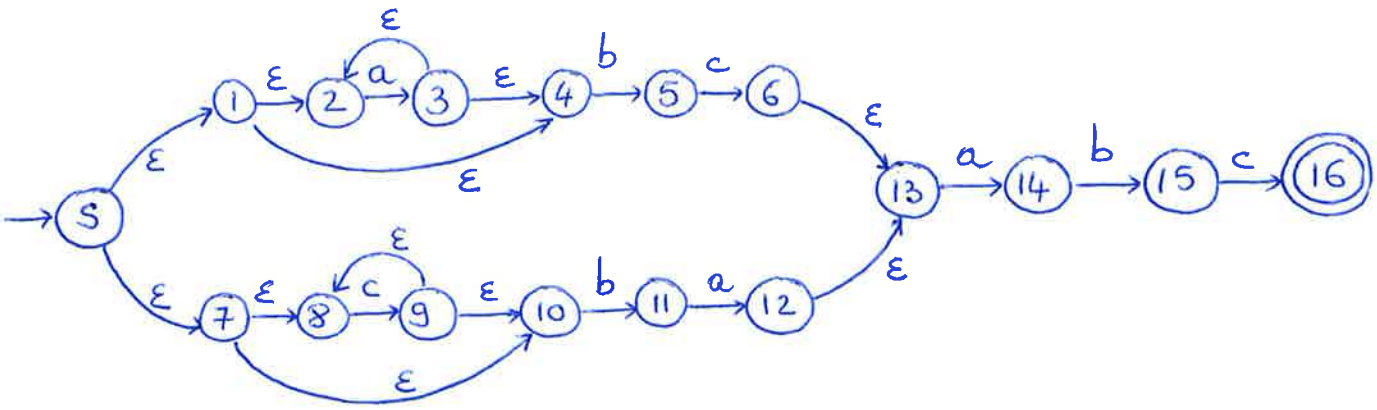
1.4 What does it mean for a given set, say S, to be closed under a given binary operation, say $\bowtie$? [10 points]

It means that for every two elements in S, say s_1 and s_2, their combination using the $\bowtie$ operator, say $s_1 \bowtie s_2$, is also part of S.

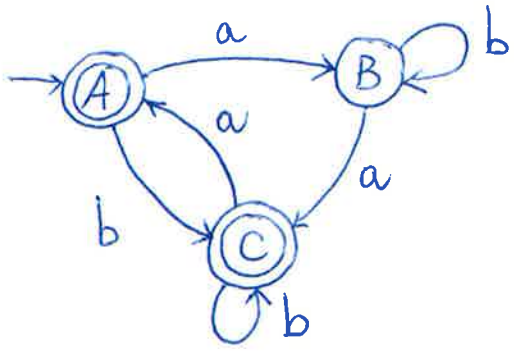
or: $\forall s_1, s_2 \in S, \; s_1 \bowtie s_2 \in S$.

2 Regular Languages [30 points]

2.1 Build an NFA – from smaller NFAs as shown in class – for the following language: $L = (a^*bc \cup c^*ba).abc$. [10 points]



2.2 Identify – using the procedure presented in class – the language recognized by the following automaton. [10 points]



$$\begin{cases} A = \varepsilon + aB + bC \\ B = bB + aC \\ C = \varepsilon + aA + bC \end{cases}$$

$$C = b^*(\varepsilon + aA)$$

$$B = bB + ab^*(\varepsilon + aA) = b^*ab^*(\varepsilon + aA)$$

$$A = \varepsilon + ab^*ab^*(\varepsilon + aA) + bb^*(\varepsilon + aA)$$

$$= \varepsilon + (ab^*)^2 + b^+ + ((ab^*)^2a + b^+a)A$$

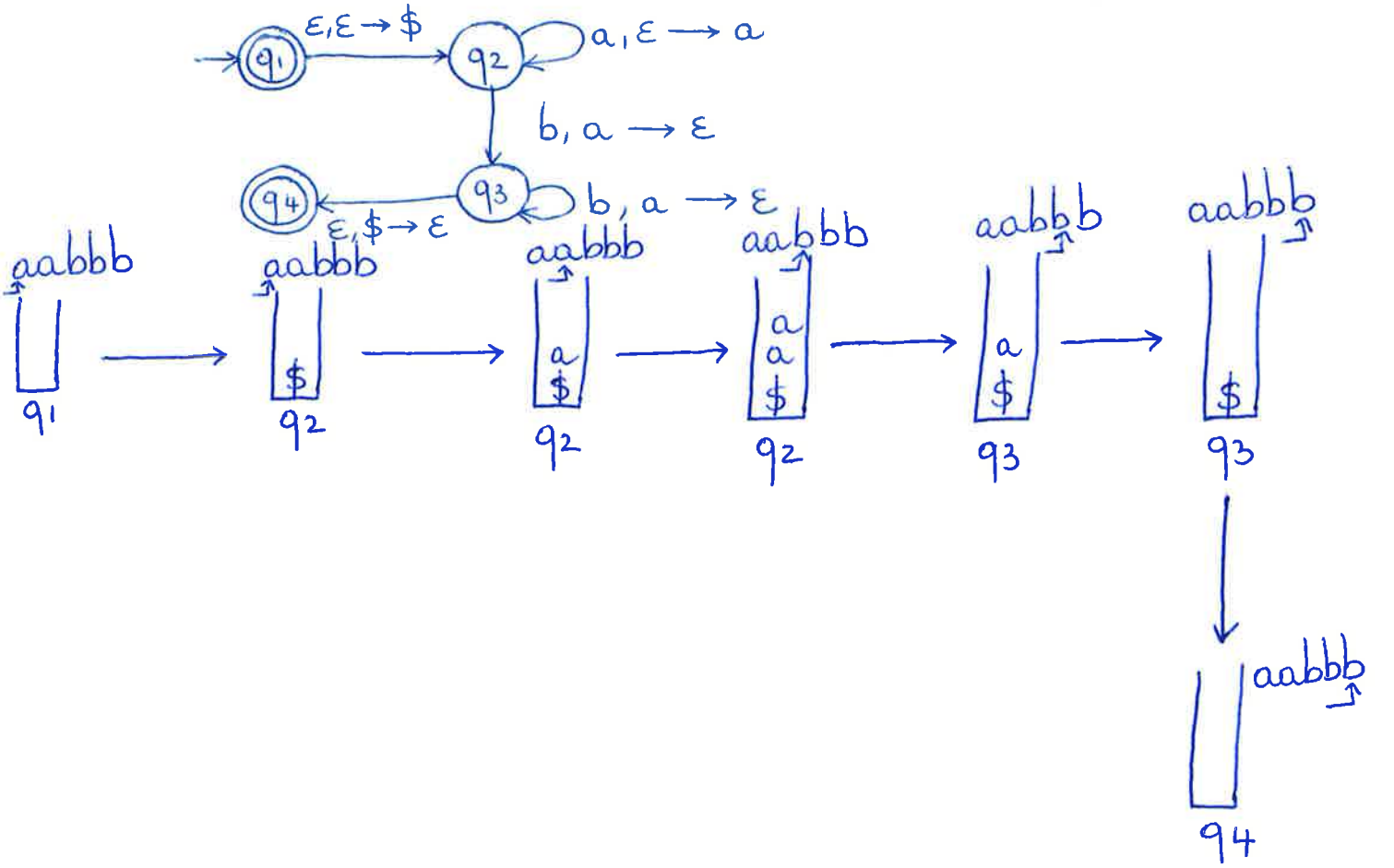
$$= ((ab^*)^2a + b^+a)^*.(\varepsilon + (ab^*)^2 + b^+)$$

2.3 **Explain mostly in plain English (possibly with the help of some diagrams if you would like) why regular languages are "pumpable". [10 points]**

see material posted on piazza.

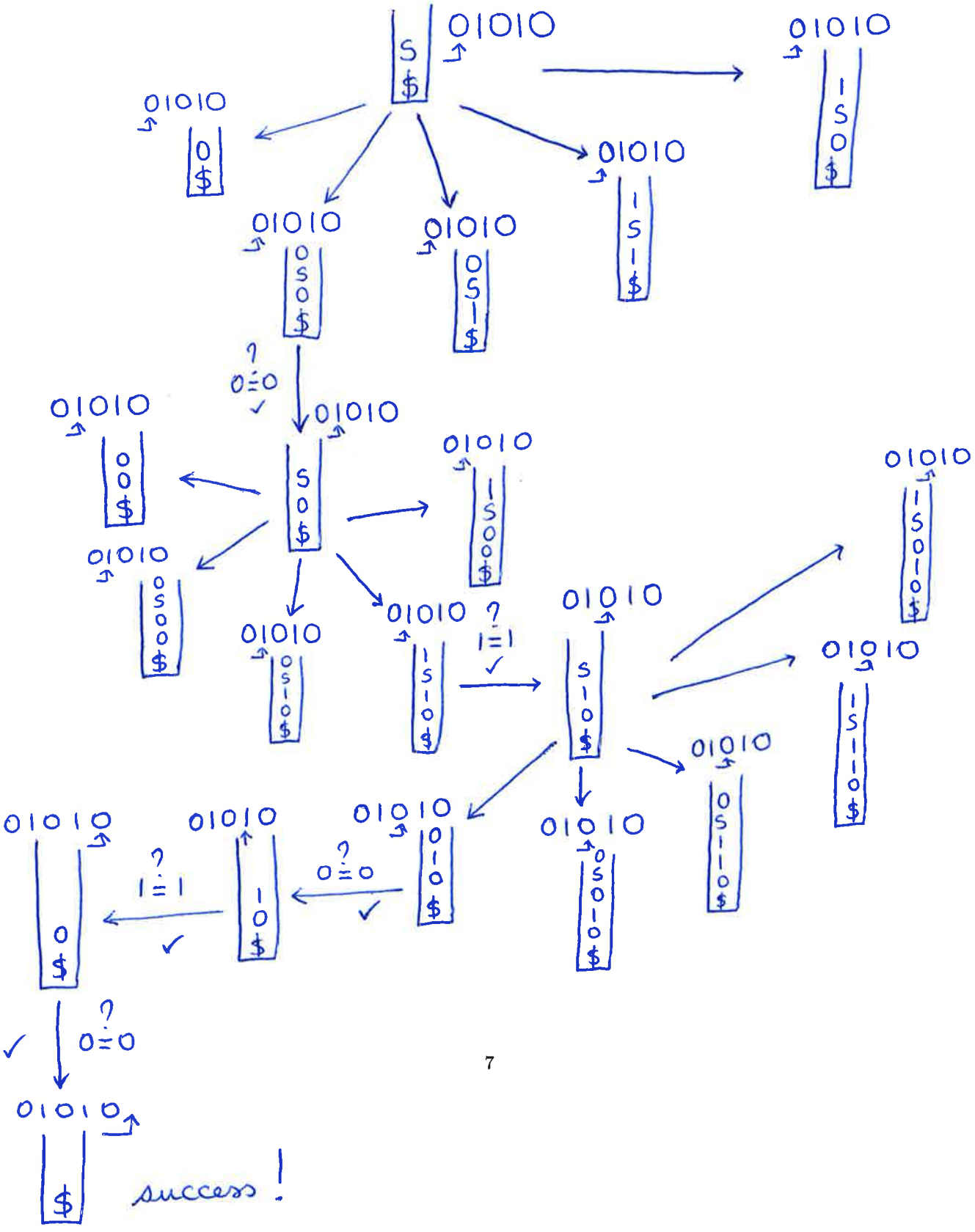
3 Context-Free Languages [30 points]

3.1 Trace the execution of the following PDA on string *aabbb*. [10 points]



3.2 **Given the following grammar, simulate the execution of the corresponding PDA on string** 01010. **[10 points]**

$S \to 0 \mid 0S0 \mid 0S1 \mid 1S1 \mid 1S0$



3.3 **Is the set of Context-Free Languages closed under concatenation? Justify your answer. [5 points]**

Any two languages L_1 and L_2 that are context free have CFGs to represent / recognize them. Let's say that: $S_1 \rightarrow \ldots$ is the CFG for L_1,

and: $S_2 \rightarrow \ldots$ is the CFG for L_2.

Then $S \rightarrow S_1 S_2$, $S_1 \rightarrow \ldots$, $S_2 \rightarrow \ldots$ is also a CFG and it recognizes $L_1 . L_2$.

Therefore $L_1 . L_2$ is a CFL.

CFLs are closed under concatenation.

3.4 **Is the set of Context-Free Languages closed under intersection? Justify your answer. [5 points]**

Let us look at the following two languages:

$L_1 = \{ a^n b^n c^k \ / \ n \geq 0, \ k \geq 0 \}$

$L_2 = \{ a^k b^n c^n \ / \ n \geq 0, \ k \geq 0 \}$

L_1 and L_2 are context-free languages.

$L_1 \cap L_2 = \{ a^n b^n c^n \ / \ n \geq 0 \}$

$L_1 \cap L_2$ is not a CFL (here: you can either use the pumping lemma for CFLs to disprove that $L_1 \cap L_2$ is a CFL, or just say that it is not a CFL because, informally, we see that there are three quantities to balance and this is beyond the expressiveness of CFLs).

Since there exist two CFLs whose intersection is not a CFL, we conclude that the set of context-free languages is not closed under intersection.

4 Turing Machines [10 points]

4.1 Give the formal definition of a Turing Machine. [5 points]

A TM is fully described by the following seven components:

- Q = set of states
- q_0 = start state
- q_{accept} = accept state
- q_{reject} = reject state
- Σ = alphabet of input ($\sqcup \notin \Sigma$)
- Γ = alphabet of tape ($\sqcup \in \Gamma$ and $\Sigma \subset \Gamma$)
- transition function δ: $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$.

4.2 List 3 differences between a Turing Machine and finite state machines we saw earlier this semester. [5 points]

- TM can move left or right on input (and tape)
- TM allow to overwrite the input
- TM have specific accept and reject states.
- a potential outcome of a TM execution is looping.