Quiz - Final Cumulative Quiz

Name ______________________________

1. [16 points] Answer the following True/False questions.

An *NFA* cannot be simulated with a *Deterministic Turing Machine* because the latter cannnot use *non-determinism*	**True**	**False**
Non-determinism creates threads of computation that run in parallel, and for both *PDA*s and *DTM*s, the memory (stack / tape) is shared across threads	**True**	**False**
*DTM*s can decide all *regular languages*	**True**	**False**
If a *DFA* has n states, and accepts one string of length n, then the language of this *DFA* is of infinite size	**True**	**False**
A *Turing Machine* might execute without reading all of the input, but a *PDA* will always read the entire input string	**True**	**False**
If a *Turing Machine* loops forever, then the input is *rejected* (though the machine never officially decides this)	**True**	**False**
Both *pumping lemmas* that we studied state that there exist strings w in the language for which all substrings of w can be pumped	**True**	**False**
For any *regular language* R, a *PDA* that never uses its stack can be developed that recognizes R	**True**	**False**
Regular expressions cannot contain variables (for example, $a^n b$ contains the variable n)	**True**	**False**
When we convert an *NFA* into a *DFA*, the resulting machine will have $O(n^2)$ states compared to the original *NFA*	**True**	**False**
A *Turing Machine* with an infinite *2D-Array* of tape (which is indexed using a two-parameter tuple (i, j)) increases the power of the traditional *TM*	**True**	**False**
*NTM*s cannot decide any additional functions over *DTM*s, but they can often make those decisions *faster*	**True**	**False**
Using a *DFA*, one can decide the language Σ^* using just one state	**True**	**False**
Using a *PDA*, one can decide the language $0^n 1^n$ using just two states	**True**	**False**
A *Turing Machine* can decide the language $0^n 1^n$ without examining every character of input	**True**	**False**
All *PDA*s MUST push a dummy symbol (such as the dollar sign) to the bottom of the stack	**True**	**False**

2. [4 points] For each machine type we have seen and proposed alteration to that machine, select whether it causes the set of recognizable functions to decrease, increase, or remain unchanged.

Machine	**Alteration**			
DFA	You have access to a tape with exactly 100 cells	**Decrease**	**Increase**	**Unchanged**
NFA	Threads can react to whether a different thread accepts/rejects (e.g., accept if thread 54 accepts)	**Decrease**	**Increase**	**Unchanged**
PDA	You can only push to the stack and read the top character (no pop)	**Decrease**	**Increase**	**Unchanged**
DTM	You are given a flag that can be read. This flag is automatically set to true if your machine will ever write a 1 to cell 10 of the tape	**Decrease**	**Increase**	**Unchanged**

3. [6 points] For each language below, choose whether it is regular, context-free, turing-decidable, turing-recognizable, or none (not even turing recognizable).

$\{0^i w \mid i \in \mathbb{N}, w \in \{0,1\}^*$, w contains at least i 0's $\}$	**Reg**	**Cont. Free**	**Tur. Dec.**	**Tur. Rec.**	**Not Tur. Rec.**
$\{a^i b^j c^k \mid i, j, k \geq 0, i = j \oplus j = k\}$	**Reg**	**Cont. Free**	**Tur. Dec.**	**Tur. Rec.**	**Not Tur. Rec.**
Is boolean formula θ *Satisfiable*?	**Reg**	**Cont. Free**	**Tur. Dec.**	**Tur. Rec.**	**Not Tur. Rec.**
Does string w contain every lower-case english letter (a-z) in order (the letters do not need to be adjacent)?	**Reg**	**Cont. Free**	**Tur. Dec.**	**Tur. Rec.**	**Not Tur. Rec.**
Given a *Turing Machine* M and input w, return true if M's head **never** reaches cell 1000 on the tape on input w.	**Reg**	**Cont. Free**	**Tur. Dec.**	**Tur. Rec.**	**Not Tur. Rec.**
$\{wtw^R \mid t \in \{0,1\}^*, \lvert w\rvert = \lvert t\rvert\}$	**Reg**	**Cont. Free**	**Tur. Dec.**	**Tur. Rec.**	**Not Tur. Rec.**

4. [3 points] Consider the language $L = 0^n 0^n 100$ where $n \geq 0, \Sigma = \{0,1\}$. First, decide if the language is regular, context-free, or Turing Decidable. In the left box, write out a regular expression, context-free grammar, or TM pseudocode for the language. In the right box below, draw a DFA, PDA, or TM given your choice.

Each of the questions below will ask you to answer a question that *relates two different topics from two different modules*. Answer each of the questions as succinctly as you are able.

5. [2 points] Are there any *regular languages* that can only be *recognized* (not decided) by a *Turing Machine*? Explain why or why not.

6. [2 points] Suppose I take a *context-free language* L and a *regular language* R. Will $L \cup R$ always be context-free? Is it possible that $L \cup R$ is regular? Explain why or why not.

7. [2 points] Suppose I have a *context-free grammar* (variables and substitution rules) G and want to know if G is an infinite language or not? Is this problem *decidable*? If so, give a brief description of how to decide it. If not, give a short explanation why.

8. [2 points] Suppose somebody proves that $P = NP$ by finding a fast (let's say $\Theta(n^3)$) algorithm for the *Traveling-Salesperson Problem*. Does that mean that all languages (functions) in *NP* are now *regular*? Explain why or why not.