1 Format

The exam will be 90 minutes (60 minutes to solve, 30 minutes to LaTeX) and will consist of TRUE/FALSE with no explanation, TRUE/FALSE with explanation, and open response problems. You are allowed to have a cheat sheet which you will have to submit with the exam.

2 Topics Covered

This is a list of some of the topics covered in the course. You may want to add some of them to your cheat sheet.

2.1 Math

- **Functions:** Know the definitions for functions, injectivity, surjectivity, bijectivity, and what these imply for the caridinality between the domain and codomain. Also know the pigeon hole principle.
- **Big-O notation:** Know the definitions for $o, O, \omega, \Omega, \Theta$ and how to identify the relation between two functions.

2.2 Data Representation

- **Representation Schemes:** Know the definition of a representation scheme (also known as an encoding), the definition of a prefix-free encoding, and ways to make any encoding prefix-free.

2.3 Circuits

- **Representation:** Know the different representations of a circuit: directed graph, straight-line program, and tuple representation.
- **Universality and EVAL:** Know that we can compute every function with a NAND-CIRC and the implications of being able to compute the EVAL functions.

  The exact implementation of the circuit that computes EVAl isn't super important for this exam.
- **SIZE(n) and Size Hierarchy Theorem:** Know the definition of $SIZE(n)$ and result of the size hierarchy theorem.
- **Comparing Languages:** Know what it means to compare the power of different languages and how to do it.

2.4 Deterministic Finite Automata & Regular Expressions

- **DFAs and NFAs:** Know the definition of a DFA, how to express one in a transition table, how to understand DFAs and NFAs, how to create one for a given language
- **Regular Expressions:** Know the definition of a regular expression, how to understand regular expressions, how to create one for a given language
- **DFA/NFA/Regex Equivalence:** Know that Regular Expressions and DFAs (and NFAs) are equivalent: for every given DFA we have a regular expression that accepts the same language, and vice-versa; also, for every NFA we can express it as a DFA.
- **Regular languages and their limitations:** Know the definition of a regular language, know that DFA and regular expressions can't compute

3 Practice Problems

Disclaimer: If some topics are covered here more than others, that doesn't mean they will be covered more or less on the midterm.

3.1 TRUE/FALSE

Write whether the following statements are true or false. No need to provide justification but you should justify it to yourself.
(*About 2 minutes each*)

1. Let $f(x) = \binom{x}{4}$ and $g(x) = \frac{2^x}{x^{10}}$.
   (a) $f = o(g)$
   (b) $f = O(g)$
   (c) $f = \theta(g)$
   (d) $f = \Omega(g)$
   (e) $f = \omega(g)$
2. The function $EQUALS : \{0,1\}^{2n} \rightarrow \{0,1\}$, which takes as input $x, x' \in \{0,1\}^n$ and outputs 1 iff $x = x'$, is in $SIZE(10n)$.

3.2 TRUE/FALSE with justification

Write whether the following statements are true or false and provide a short justification.
(*About 4 minutes each*)

1. Consider two functions f, g. If $f = O(g)$ then $f \neq \Omega(g)$.

2. The set of circuits made from NOT and OR gates universal.

3. Let $f(x) = \binom{x}{4}$ and $g(x) = x^4 - 2x^3 + 3x^2 + 1$.

   (a) $f = o(g)$
   (b) $f = O(g)$
   (c) $f = \theta(g)$
   (d) $f = \Omega(g)$
   (e) $f = \omega(g)$

3.3 Short Answer

1. Prove or Disprove: There exists a regular expression that computes the function that returns 1 on the binary string $x \in \{0,1\}^*$ if and only if x has strictly more 1s than 0s.

2. Create an encoding function $E : DFA_n \to \{0,1\}^{10n^2}$ (for every sufficiently large n) where DFA_n is the set of DFAs with n states.