The second test for this course will be given in class on Wednesday, April 14. The test covers all of the material that we have studied from Chapter 3 and Sections 4.1 through 4.3 in Chapter 4, except for Sections 3.3 and 3.7. (This means that you will not be asked about grep, or about other Linux commands, or about the complex syntax that is used for practical regular expressions. And there will be no Pumping Lemma proof. Note that pushdown automata, Section 4.4, are not included.)

The test will include some "short essay" questions that ask you to define something, or discuss something, or explain something, and so on. Other than that, you can expect most of the questions to be similar to problems that have been given on the homework. There might be some simple proofs.

Here are some terms and ideas that you should be familiar with for the test:

alphabet (finite, non-empty set of "symbols")
string over an alphabet Σ
length of a string, $|x|$
empty string, ε
concatenation of strings, xy or $x \cdot y$
reverse of a string, x^R
x^n, for a string x and a natural number n
$n_\sigma(x)$, the number of occurrences of a symbol σ in a string x
the set of all possible strings over Σ, denoted Σ^*
language over an alphabet Σ (a subset of Σ^*)
a language over Σ is an element of $\mathscr{P}(\Sigma^*)$
the set of strings over Σ is countable; the set of languages over Σ is uncountable
union, intersection, set difference, and complement applied to languages
concatenation of languages: LM, L^n for $n \in \mathbb{N}$
Kleene-star operation on a language: L^*
regular experssion over an alphabet Σ; the operators: `*`, `|`, and concatenation
regular language; the language $L(r)$ generated by a regular expression r
DFA (Deterministic Finite Automaton)
transition diagram [the usual picture] of a DFA
state (in a finite-state automaton); start state; accepting state (also known as final state)
definition of a DFA as a list of five things, $(Q, \Sigma, q_o, \delta, F)$ — and what each thing means
how a DFA computes (that is, what it does when it reads and processes a string)
NFA (Non-deterministic Finite Automaton); the differences between NFAs and DFAs
nondeterminism
ε-transitions
what it means for an NFA to accept a string
the language, $L(M)$, accepted by an NFA or DFA, M
algorithm for converting an NFA to an equivalent DFA

algorithm for converting a regular expression to an equivalent NFA
DFAs, NFAs, and regular expressions all define the same class of languages
operations ($L \cup M$, $L \cap M$, LM, L^*, $\overline{L}$, L^R) on regular languages produce regular languages
CFGs (Context-Free Grammars)
production rules; non-terminal and terminal symbols; start symbol
definition of a CFG as a list of 4 things, $G = (V, \Sigma, P, S)$
derivation (of a string from the start symbol of a CFG)
the language, $L(G)$, generated by a CFG, G; context-free language
if L and M are context-free languages, then so are $L \cup M$, LM, and L^*
every regular language is context-free
BNF (Backus-Naur Form)
using BNF to define the syntax of a language
parsing
parse tree
left derivations and right derivations
how a parse tree coressponds to a derivation

examples of languages that are not regular, such as:

$\{a^n b^n \mid n = m\}$
$\{a^n b^m c^k \mid k = n + m\}$
$\{w \in \{a, b\}^* \mid w = w^R\}$
$\{w \in \{a, b\}^* \mid n_a(w) < n_b(w)\}$
$\{ww \mid w \in \{a, b\}^*\}$
$\{a^n b^n c^n \mid n \in \mathbb{N}\}$
$\{a^{n^2} \mid n \in \mathbb{N}\}$
$\{www \mid w \in \{a, b\}^*\}$

examples of languages that are context-free but not regular, such as:

the **first four** languages in the previous list

some of the tasks that you could be asked to perform:

finding a regular expression for a given language
finding a DFA or NFA for a given language
finding a regular expression for an NFA or DFA
converting an NFA into an equivalent DFA, using the algorithm
converting a regular expression into an equivalent NFA, using the algorithm
determining whether a given string is accepted by an NFA or DFA
finding a derivation for a given string from a given CFG
finding a CFG for a given language
finding the language generated by a given CFG
using BNF in basic ways
finding a parse tree for a given string from a given CFG
finding a left derivation or a right derivation

Here are some practice problems from old exams. . .

*Note that this is **not** meant to be a complete review of everything that might be on the test. It is meant to give you a feeling for what kinds of test questions are possible*

1. a) Suppose that L and M are languages over an alphabet Σ. How is the language LM defined?

b) Suppose that $\Sigma = \{a, b, c\}$ and that L is the language $L = \{a, b\}$ and M is the language $M = \{w \in \Sigma^* \mid w \text{ ends with a } c\}$. What strings are in the language LM ? (Describe the language with words, **not** with a regular expression.)

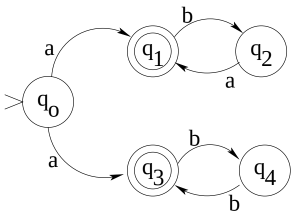
2. Consider the following context-free grammars:

$$\begin{array}{llll} S \longrightarrow TR & \qquad\qquad & S \longrightarrow SS \\ T \longrightarrow aTb & & S \longrightarrow T \\ T \longrightarrow aT & & T \longrightarrow aTb \\ T \longrightarrow a & & T \longrightarrow \varepsilon \\ R \longrightarrow aRb & & \\ R \longrightarrow bR & & \\ R \longrightarrow b & & \end{array}$$

a) For the grammar on the **left**, find a *left derivation* and the corresponding *parse tree* for the string *aaabbabbb*.

b) For the grammar on the **right**, find **two** *left derivations* and the two corresponding *parse trees* for the string *abaabbab*.

3. Consider the following Nondeterministic Finite Automaton:



a) Give a regular expression for the language accepted by this NFA.

b) Apply the NFA-to-DFA conversion algorithm to construct a DFA that accepts the same language as this NFA.

4. Consider the regular language $L = \{w \in \{a, b\}^* \mid w \text{ contains the substring } abaab\}$. Draw a DFA that accepts exactly this language.

5. Give a regular expression that generates each language (no explanation necessary):

$$L_1 = \{w \in \{a, b\}^* \mid w \text{ contains } \textbf{at least 2 } a\text{'s}\}$$
$$L_2 = \{w \in \{a, b\}^* \mid w \text{ contains } \textbf{exactly 2 } a\text{'s}\}$$

6. Draw an NFA that accepts the language over the alphabet $\Sigma = \{a, b, c\}$ that is generated by the regular expression $(a|b)^*cc^*(a|b)^*$. You do **not** have to use a specific regular-expression-to-NFA conversion algorithm; any NFA that works will do.

7. Let L be a language over some alphabet Σ. Suppose that $\varepsilon \in L$. Show that $L \subseteq L^2$.

8. What does it mean that $x^R = x$ for a string x? Let $\Sigma = \{a, b, c\}$. Give several examples of strings, x, over the alphabet Σ that have the property that $x^R = x$, and explain in English what this property means.

9. Give a Context-Free Grammar for the language $L = \{a^n b^m c\, b^m a^n \mid n \in \mathbb{N},\ m \in \mathbb{N}\}$, and briefly explain how your grammar works.

10. ("$a^n b^m$ variations") Match each language with the grammar that generates it. Enter the number for the correct grammer. No explanation is necessary.

a) $\{a^n b^m \mid n \neq m\}$ Grammar # ____

b) $\{a^n b^m \mid n \geq m\}$ Grammar # ____

c) $\{a^n b^m \mid n > 2m\}$ Grammar # ____

d) $\{a^n b^m \mid n > 0\}$ Grammar # ____

e) $\{a^n b^m \mid n > m > 0\}$ Grammar # ____

f) $\{a^n b^m \mid n = 2m + 1\}$ Grammar # ____

---

1)
$S \longrightarrow aSB$
$S \longrightarrow \varepsilon$
$B \longrightarrow b$
$B \longrightarrow \varepsilon$

2)
$S \longrightarrow aS$
$S \longrightarrow Sb$
$S \longrightarrow a$

3)
$S \longrightarrow aaSb$
$S \longrightarrow aT$
$T \longrightarrow aT$
$T \longrightarrow \varepsilon$

4)
$S \longrightarrow aaTb$
$T \longrightarrow aTb$
$T \longrightarrow aT$
$T \longrightarrow \varepsilon$

5)
$S \longrightarrow aSb$
$S \longrightarrow T$
$S \longrightarrow R$
$T \longrightarrow aT$
$R \longrightarrow Rb$
$T \longrightarrow a$
$R \longrightarrow b$

6)
$S \longrightarrow aaSb$
$S \longrightarrow a$

11. Let $G = (V, \Sigma, P, S)$ be a Context-Free Grammar. $L(G)$ is the language generated by G. How is $L(G)$ defined? That is, exactly what does it mean for a string, w, to be in $L(G)$? (Mention Σ, P, and S in your answer.)

12. NFA stands for "Nondeterministic Finite Automaton." Discuss how an NFA works and what it means to say that it is "nondeterministic." (What does an NFA do as it reads an input string? What does it mean to say that the NFA accepts the string?)