Composite Sample of CS81 Exam Problems (from midterms or finals)

---

1. Classify these formulas as one of
   {tautology, satisfiable but not a tautology, not satisfiable}:

   a. $((p \rightarrow q) \rightarrow (q \rightarrow r)) \rightarrow (p \rightarrow r)$
   b. $(p \rightarrow q) \rightarrow (q \rightarrow p)$
   c. $\neg(\neg q \rightarrow p)) \rightarrow \neg(p \wedge \neg q)$
   d. $(p \wedge (q \vee r)) \equiv ((p \wedge q) \vee (p \wedge r))$

2. For each formula above:

   a. If the formula is a tautology, give a natural deduction proof of it.
   b. If the formula is unsatisfiable, give a natural deduction proof of its negation.
   c. If the formula is satisfiable, but not a tautology, give an assignment that satisfies it and one that does not.

[10 points]

a. [3/10 points]

Give a context-free grammar in Chomsky normal form for the well-balanced parentheses language with two kinds of parentheses: [ ] and ( ), excluding the empty string. For example, the following string is in the language: [([ ] ( ))] ([( ) [ ]]).

b. [7/10 points]
For each of the following two strings: If the string is in the language of your grammar, prove it by constructing a derivation tree for it. If it is not in the language, prove it by giving the table from the CYK algorithm:

i. ([ ] ([ ] [ ]])
ii. ([ ] ([ ]) [ ])

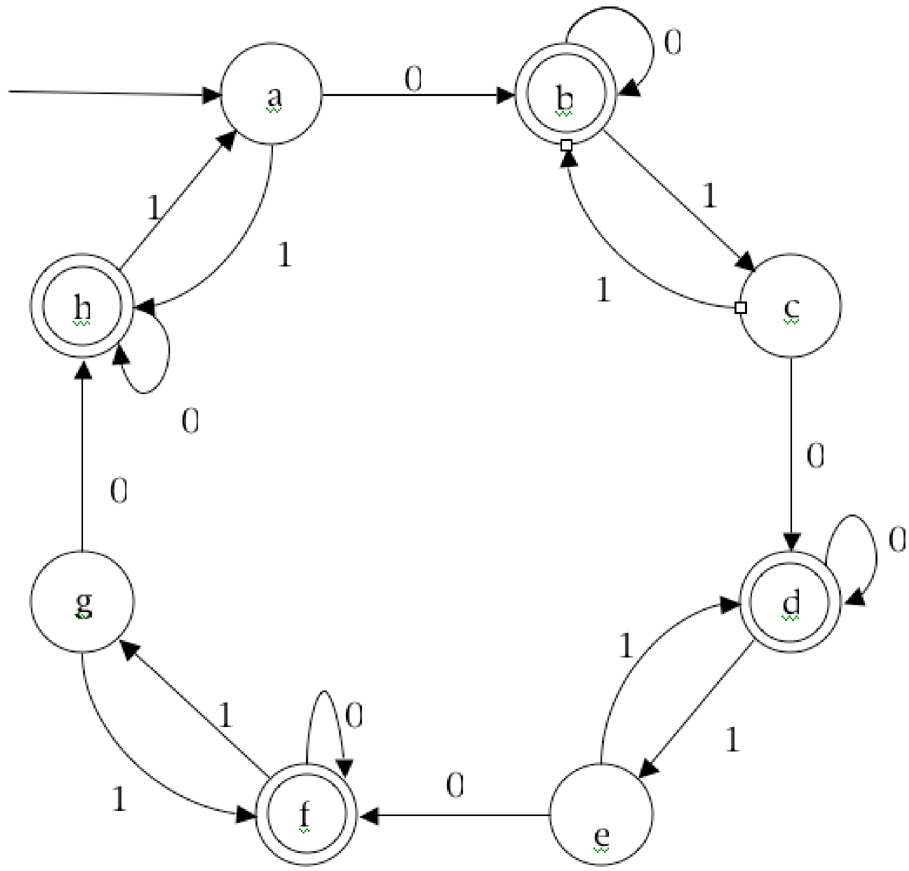
[5 points]

Using natural deduction, derive the rule known as *modus tollendo ponens* from the basic rules given in the text:

$$\frac{\varphi \vee \psi, \quad \neg\varphi}{\psi}$$

[10 points]

Give a regular expression for the language accepted by the following DFA:



[10 points]

Give the transition diagram for a pushdown acceptor that accepts the language

$\{x \in \{a, b, c\}^* \mid \#_a(x) = \#_b(x) + \#_c(x)\}$

For example, abac is in the language, but abc is not.

[20 points]

Consider the following language categories:

- 4: Finite language
- 3: Regular language, but not finite
- 2: Context-Free Language, but not regular
- 1: Not context-free

For each language in the following table, indicate the category in which the language belongs.

	Language	Category
a.	$\{0^n1^n0^n1^n \mid n \in \omega\}$	
b.	$\{0^n1^n \mid n \in \omega\}\{0^n1^n \mid n \in \omega\}$	
c.	$\{1^{1000n} \mid n \in \omega\}$	
d.	$\{0^{1000}, 1^{1000}\}$	
e.	$\{0\}^*\{0^n1^n \mid n \in \omega\}\{1\}^*$	
f.	$\{0^n1^n \mid n \in \omega\}^*$	
g.	$\{0^n10^n \mid n \in \omega\}$	
h.	$\{0^n0^n \mid n \in \omega\}$	
i.	$\{xxx \mid x \in \{1\}^+\}$	
j.	$\{xy \mid x, y \in \{0, 1\}^+, \lvert x\rvert = \lvert y\rvert\}$	

[10 points]

Construct a minimum-state deterministic finite-state acceptor that accepts all strings in {0, 1}* not containing 0101.

[10 points]

Use the pumping lemma to show that the language

$$\{a^ib^{i*i} \mid i \in \omega\} = \{\Lambda, ab, aabbbb, aaabbbbbbbbb, \ldots\}$$

is not context-free.

[10 points]

The "Regular Expression Substitution Principle" says that for any regular expression equality we can substitute entire languages in place of the letters and the equality remains valid.

For example, given the equality (0*1)* = (1*0)*, we infer the equality (L*M)* = (M*L)* for any languages L and M.

Assuming that this principle is correct, describe an algorithm for testing the validity of equalities for arbitrary languages, where the equalities involve only the operators union, language product, and star.