For each of the following languages, decide if they are CFL, if they are then provide a CFG; if not then use the pumping lemma to show that they are not CFL

(a) the set of strings over the alphabet $\{a, b\}$ with more a's than b's

(b) $\{ w\#x \mid w^R$ is a substring of x for $w, x \in \{0,1\}^* \}$

(c) Let $\Sigma = \{1,2,3,4\}$ and $C = \{ w \mid w \in \Sigma^*$ and in w, the number of 1s equals the number of 2, and the number of 3s equals the number of 4s$\}$

(c) (1) Assume that C is a CFL

(2) let p be the pumping length that is defined by the pumping lemma

(3) Let $S = 1^p 3^p 2^p 4^p$

$|S| = 4p \quad S \in C$

(4) S may be divided into five parts

$S = u v^i x y^i z \;\; \forall i \geq 0 \;\; S \in C$

$|vy| > 0$

$|vxy| \leq p$

if $u = \varepsilon$ vxy contains at least one 1 and for $i > 1$ then S contains more one's than twos' ~~threes fours~~

(5) We found a contradiction $S \notin C$ for $i > 1$; therefore C is not CFL

Name Quevedo

Computational Models
Quiz 4

1. The following grammar generates all regular expressions over the alphabet {a,b,c}:

   rexp-> rexp **U** *rexp* | *rexp rexp* | *rexp* * | **(** *rexp* **)** | *a* | *b* | *c*

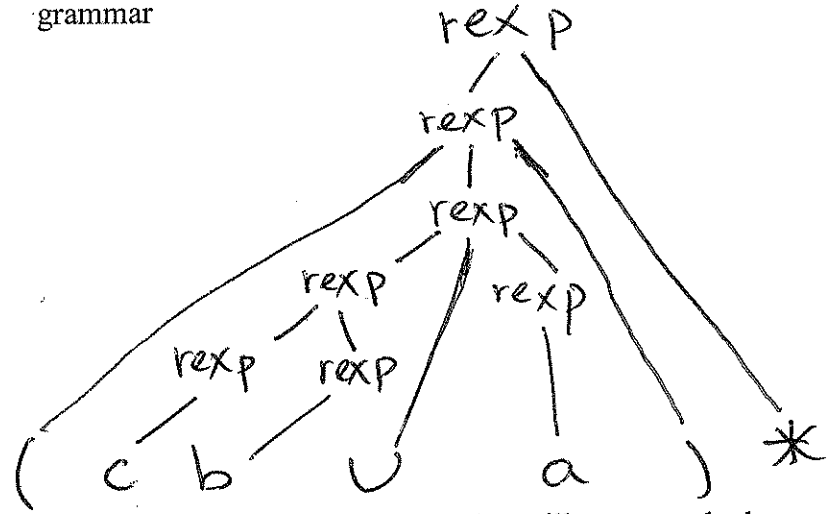
   a. *(5 points)* List the variables of the grammar

   { rexp }

   b. *(5 points)* List the terminals of the grammar

   { U , * , (,) , a , b , c }

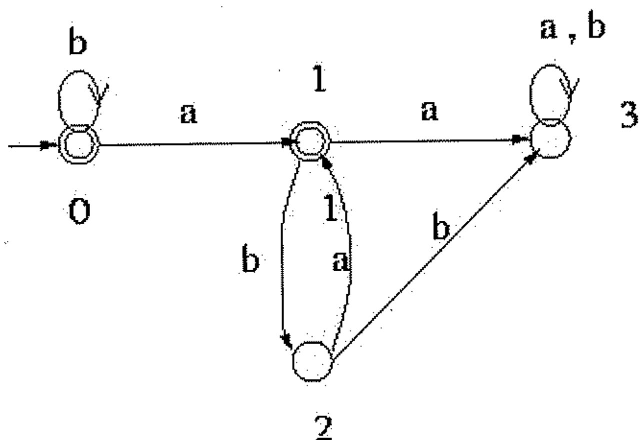
   c. *(30 points)* Draw a parse tree for the regular expression (cb U a)* using this grammar



2. *(30 points)* Find a context free grammar that will generate the language (ab U cb)*

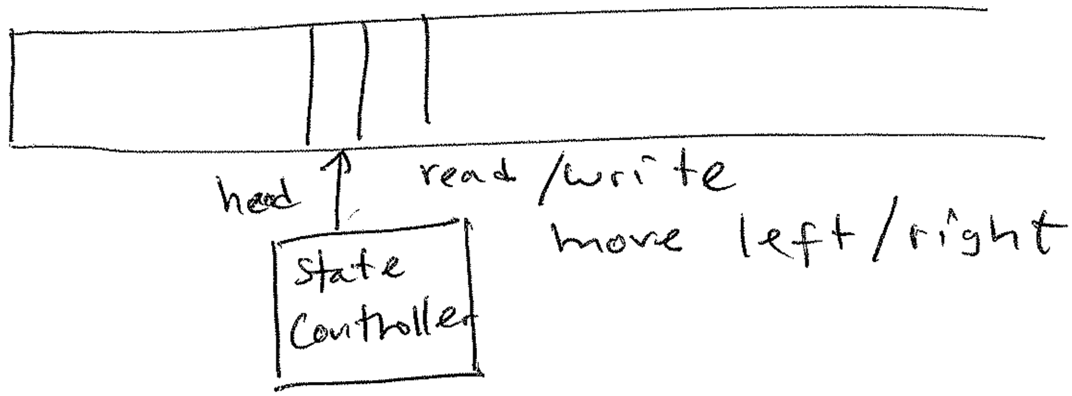
   S → abS | cbS | ε

3. *(30 points)* Find a context free grammar that will generate the language accepted by the following FA. Use procedure described in class.



0 → a1 | b0 | ε
1 → a3 | b2 | ε
2 → a1 | b3
3 → a3 | b3

Turing Machines



Alan Turing showed that even a TM could not solve some problems.

A TM use a tape as its infinite memory.

The TM has a tape head that can read and write symbols and move around on the tape.

1. Initially, the tape contains just the input string, and it is blank everywhere else.
2. To store information, the TM writes to the tape.
3. To read information, the TM can move its head over the tape

(4) the TM continues computing until it decides to produce an output (accept / reject) or it never halts.

Formal Definition of a TM

A Turing Machine (TM) is a 7-tuple

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

1. Q is a set of states
2. Σ is the input alphabet, not including the special blank symbol $\sqcup$
   bl
3. Γ is the tape alphabet.
   $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
4. $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
5. $q_0 \in Q$ is the start state
6. $q_{accept} \in Q$ is the accept state
7. $q_{reject} \in Q$ is the reject state

NAME

Computational Models
In class assignment

Team practice

1. Using the information provided in class and from your textbook pages 140-143, solve the following questions.
   a. If M is a Turing machine, what is L(M)?

      Language recognized by M

   b. What is a *configuration* of the TM?

      current state, tape contents and current head location

   c. How do we call the behavior that may never lead a TM to a halting state?

      looping

   d. What is a "*recursively enumerable language*"?

      language that a TM recognizes

   e. What is a "*decidable language*"?

      language that a TM decides

   f. What is a "*recursive language*"?

      same as in (e)

   g. What is a "*Turing-recognizable language*"?

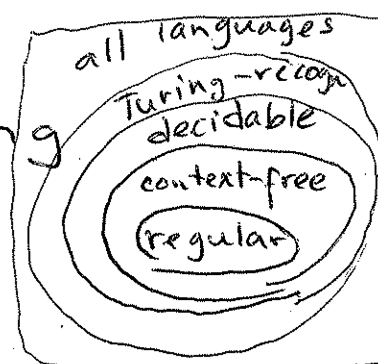
      same as in (d)

   h. Which configurations of a TM are *halting*?

      accepting and rejecting

   i. Draw a Venn diagram that illustrates the relationship among *Turing-recognizable*, *context-free*, *decidable*, *regular*, and *all* languages.

      see figure 4.10 page 173



   j. Are all TM *deciders*?

      No

2. Examine the formal definition of a Turing Machine from page 140 to answer the following questions, and be ready to explain your reasoning.

   See exercise 3.5 page 160 and 162

   a. Can a TM ever write the blank symbol on its tape?

      yes

   b. Can the tape alphabet be the same as the input alphabet?

      no

   c. Can a TM's head ever be in the same location in two successive steps?

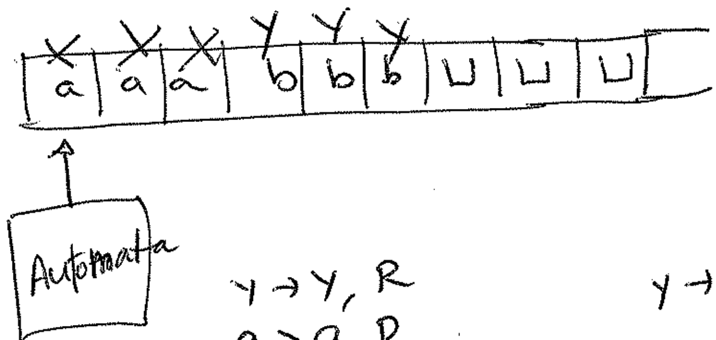
      yes

   d. Can a TM contain just a single state?

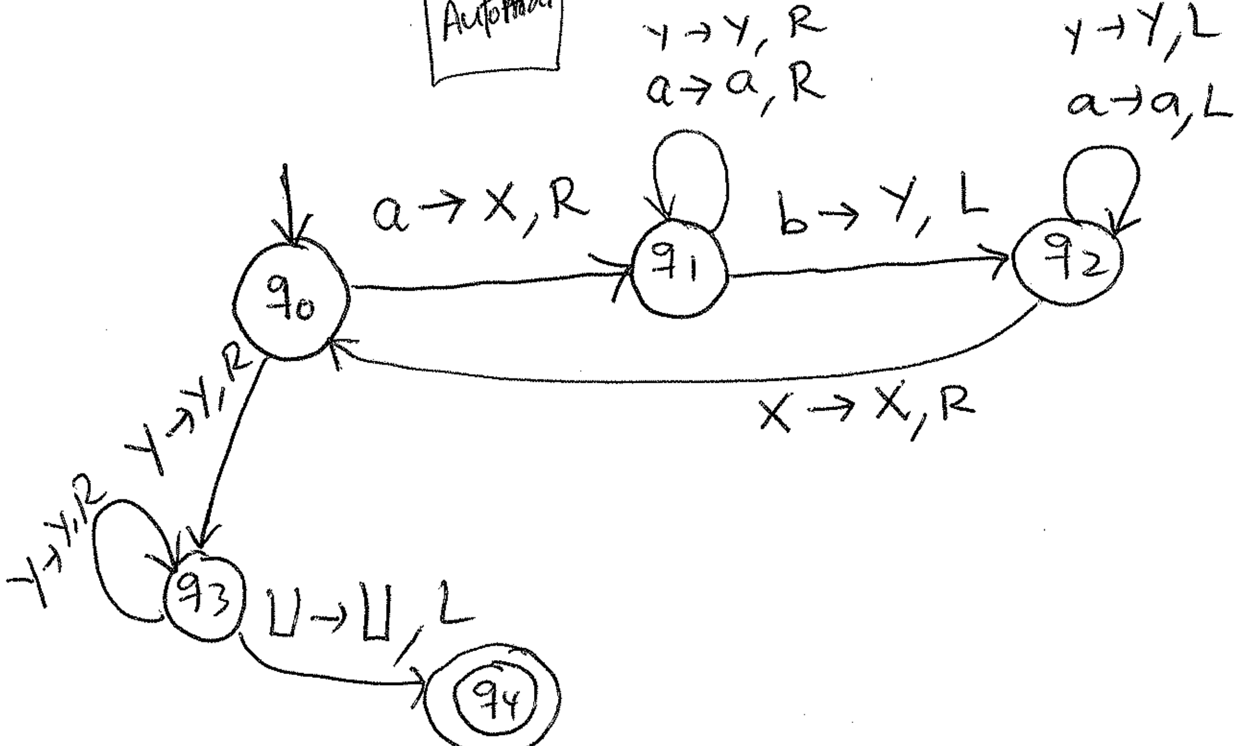
      no

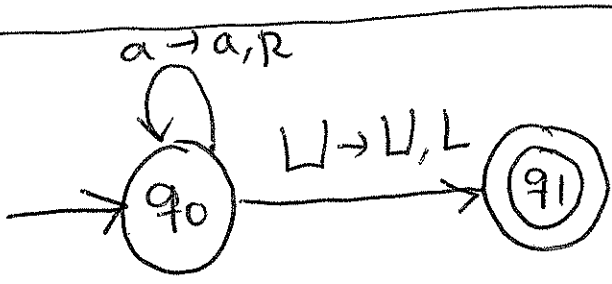
Formal Definition of a TM

Design a TM for language $\{a^n b^n \mid n \geq 1\}$



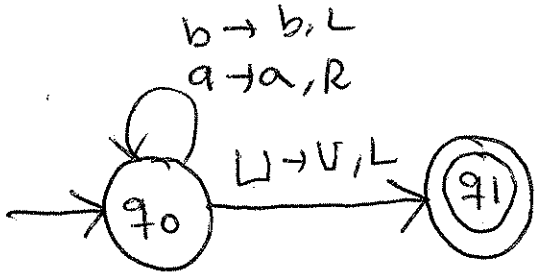
â
á
ⓐ
X
Y





Is the string aaa accepted? yes

Is the string aba accepted? no



Is the string aba accepted?

- the final state cannot be reached
- the machine never halts.
- The input is not accepted.