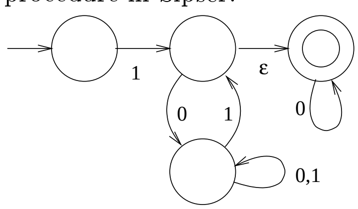
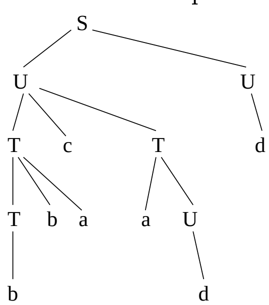
CS172, Midterm 1 Solutions

alexf@csua.berkeley.edu

1. a. Here is a simple NFA that recognizes the language requested; there are several other options as well, include a much bigger NFA produced by the formal regular expression to NFA conversion procedure in Sipser.



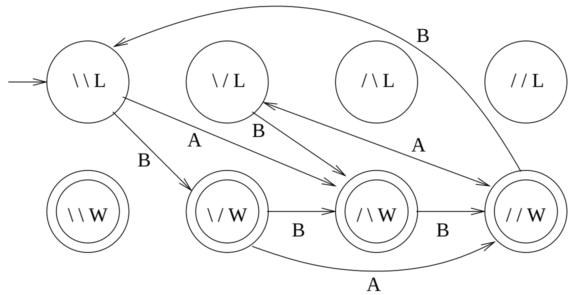
   b. This is the sole possible parse tree, quickly found by noticing that the only source of d's is U.



2. (True/False)

False	If the language L is regular, so is any subset of L. *Note that* **any** *language is a subset of* Σ^**, a regular language, and we've certainly seen that there exist non-regular languages.*
True	The regular expression $(0 \cup 1 \cup \emptyset) \circ (\epsilon \circ \emptyset)$ defines the empty language. *Examining the definition of the* $\circ$ *operation will show that, for any regular expression* R*,* $R \circ \emptyset = \emptyset$.
False	There exists an integer N such that the language P_N of all prime factors of N expressed in unary, is not regular. *No matter how large* N *is, it's still finite, and has a finite number of factors. Any finite language is regular.*
False	A Turing machine may never write a space to its tape. *There's nothing in the definition that would impose such an artificial restriction.*
True	Over a given alphabet Σ, there is only a finite number of languages accepted by a 1-state NFA. *If the sole state is not accepting, then the NFA* **must** *be accepting the empty language. If the sole state is accepting, then the language the NFA accepts is uniquely determined by the subset of* $S \in \Sigma$ *for which there are loops; the NFA then accepts* S^**. And there's only a finite number of subsets of* Σ.

3. a. Note that the "state" of the DFA should keep track not only of the positions of the two levers, but also of whether the last marble came out of the W slot, making the state an accepting state. The transitions are then directly derivable from the toy's operation rules; the figure below only shows transitions from states reachable from the starting state (the other states can be left out of the DFA). Position of the x lever is listed first in the state names.



b. Important to the description of the language accepted is the quantity $X = A + \lfloor B/2 \rfloor$, where A and B are the numbers of A and B marbles dropped in thus far, respectively. It is easily seen that the x lever is rotated X times. The actual language accepted is the union of (1) the set of all strings such that the value of X before the last marble is dropped in is even, and (2) the set of all strings ending with B, whose total count of B's is odd. The first set includes all strings which result in the last marble being directed toward W by the x switch, and the second – by the y switch.

An alternative interpretation of this language, noticed independently by Jack Sampson and Billy Chen, is to consider the toy as a 2-bit adder, started at 0, with each B marble resulting in a 1 being added to the total, and each A corresponding to a 2. Then, the machine only if the last marble dropped did *not* cause the adder to overflow.

4. Suppose there exists a p satisfying the conditions of the pumping lemma for context-free languages. Then, consider $s = a^p b^p c^p$. For any u, v, x, y, z satisfying the pumping lemma conditions — $s = uvxyz$, $|vxy| \leq p$, $|vy| > 0$ — we know, from the last condition, that vy must contain at least one symbol from the last condition. However, the second condition guarantees that the end of y (in s) is at most p symbols after the beginning of v (in s), so it cannot be the case that both an a and a c is contained in vy. Thus, uv^2xy^2z has at least $p+1$ of at least one of the symbols, but exactly p of one of the others, so it can't possibly be in the language. Thus, the pumping lemma is violated, and the language is not context-free. $\Box$

5. Since L and $\overline{L}$ are enumerable, there exist, respectively, TMs M_1 and M_2 recognizing them. A decider TM for L can be constructed by, given any input w, running M_1 and M_2 in parallel on two independent copies of w (on 2 different tapes), one step at a time. If M_1 accepts, accept; if M_2 accepts, reject. Clearly, the set of strings recognized by this TM is L, since the only strings accepted are those accepted in finite time by M_1. Furthermore, this TM is a decider since any $w \notin L$ is in $\overline{L}$ and will thus be accepted in finite time by M_2, and duly rejected by this TM.