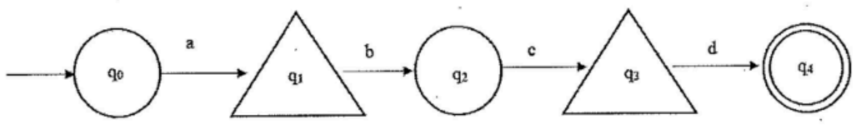
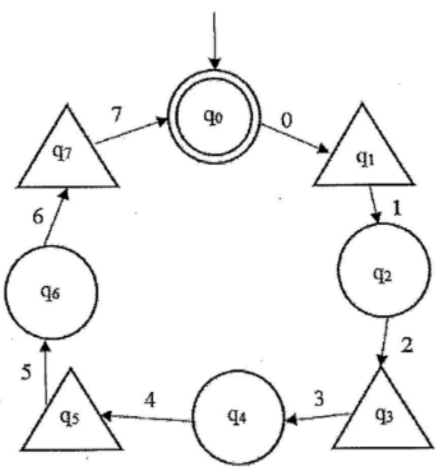
Problem 1. A *Flip-DFA (FDFA)* is like a DFA except that some of its states are flipping states. When an FDFA enters a flipping state, it reverses the remaining input (e.g. abcde becomes edcba) and then carries on computation as usual. Flipping states are indicated by being triangular. Any state may be a flipping state, including the starting state and accepting states.

Example: Consider the following FDFA F where q_1 and q_3 are flipping states, and all transitions not specified go to some other non-accepting state. On input *adcb*, F will first consume the *a* to go from state q_0 to q_1, leaving remaining input *dcb*. Then, upon entering flipping state q_1, F will reverse the remaining input to be $(dcb)^R = bcd$. F will then consume b and c to go from q_1 to q_2 and then from q_2 to q_3, leaving input *d*. After entering q_3, F will then reverse the remaining input to $d^R = d$. Then, F will consume *d* to go from q_3 to q_4 where F will accept.



a) **(15 points).** Create a FDFA that decides $L_{EQ} = \{0^n 1^n \mid n \geq 0\}$.

b) **(10 points).** Let F be the following FDFA over the alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7\}$. We assume that all transitions not specified go to some other non-accepting state. Here, q_1, q_3, q_5, and q_7 are flipping states. Give two strings other than the empty string that are accepted by F.



We will prove the following pumping lemma in two parts:

Lemma 1. *For all FDFAs F, there exists an $n \in \mathbb{N}$ such that for all $x \in L(F)$ with $|x| \geq n$, there exist strings a, b, c, d, e such that $x = abcde$, and*

i. $\forall i,\ ab^i cd^i e \in L(F)$

ii. $|b| + |d| > 0$

iii. $|abde| \leq n$

c) **(30 points).** Let an Even-Flip-FDFA F be a FDFA where every cycle in the state diagram of F has an even number of flipping states. Prove that **Lemma 1** holds for all Even-Flip FDFAs. *Hint: Part (1b) should help.*

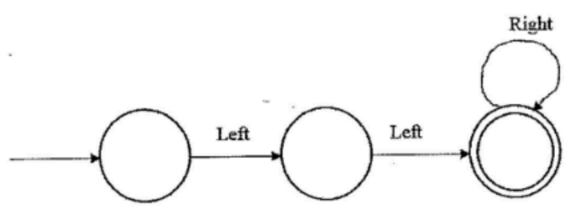
d) **(20 points).** Prove **Lemma 1** holds for *all* FDFAs.
Note that proving (1d) also proves (1c).

Problem 2. A *Nondeterministic Line Robot (NLR)* is a robot that moves along a number line. It is controlled by a hardwired program which is like an NFA except that each of its transitions require no input, take one time step to carry out, and are accompanied by a direction in $\{Left, Right\}$ indicating the direction that the robot should move on that time step. An NLR is specified solely by this control NFA.

An NLR operates as follows: It is placed at the origin of an infinite number line and powered on, putting it in its start state. At each time step while it is powered on, it must nondeterministically choose to follow one of the transitions from its current state. It will then move one unit in the direction indicated by the label on the transition, either $Left$ or $Right$. After moving, if it reaches an accepting state from this transition, then it may nondeterministically choose to power down. Alternatively, it can choose not to power down and to continue operation. If there is a case where the robot cannot transition or power down, then it self-destructs and explodes.

The location-set of an NLR R, denoted by $Loc(R)$, is the set of all positions on the number line where the robot R might end up when it powers down. Note that self-destructing and exploding at a position does not count as powering down at that position. Formally, $Loc(R) = \{x \mid R \text{ could power down at position } x\}$.

Example: The location-set generated by the NLR specified below is $\{x \mid x \geq -2\}$:



a) **(15 points).** Given two fixed numbers $a, b \in \mathbb{N}$, specify a NLR for the location-set $L = \{ax + by \mid x, y \in \mathbb{Z}\}$.

b) **(30 points).** We will prove the following theorem in two parts

Theorem 1. *Let an Unproductive-NLR be a NLR R such that every cycle in the state diagram of R has an equal number of right and left transitions. If R is an Unproductive-NLR, then Loc(R) is finite.*

i. **(20 points).** Prove that if R is an Unproductive-NLR and there is a path through R that directs R to position k in m time steps, then there exists some path to position k with no repeated states.

ii. **(10 points).** Prove **Theorem 1** using (2b.i).

c) **(30 points).** Prove that for any NLR R, there exists an $n \in \mathbb{N}$ such that if R ever powers down at a position farther than n units from the origin, then there are an infinite set of positions at which R can power down. In other words, prove that for any NLR R, there exists an $n \in \mathbb{N}$ such that if there is a $k \in Loc(R)$ with $|k| > n$, then $Loc(R)$ is infinite. *Hint: Think of how we proved the pumping lemma.*

d) **Extra Credit (30 points).** An NLR that becomes Drone-Friended[1] operates the same as before except that when it reaches an accepting state, in addition to the option of doing nothing or powering down, it can instead choose to call its drone friend to immediately fly it back to the origin. Once back at the origin, it will continue operating as usual. Note that it will continue operating from the state that it was in when it called its drone friend, and will *not* reset its state to the starting state.

Let R be an NLR such that Loc(R) is of finite size n. If R becomes Drone-Friended, then what is the maximum size of the new location-set it now recognizes? Why? Give the answer as a function of n or list it as infinity. Then explain why.