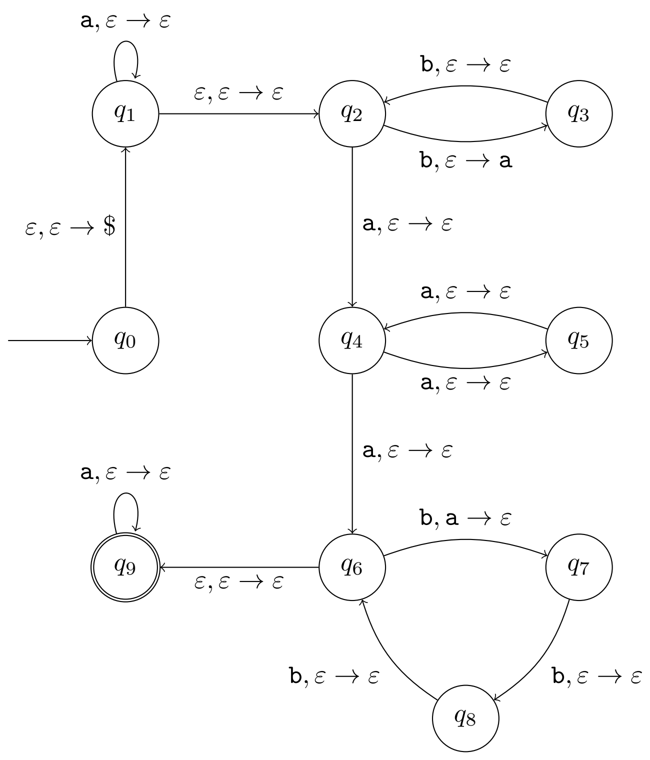
Problem 1 (150 pts)**.**

For the following PDA, give the language L_1 it accepts and a CFG G for which $L_1 = L(G)$.


$\mathtt{a}, \varepsilon \to \varepsilon$
q_1
$\varepsilon, \varepsilon \to \varepsilon$
$\mathtt{b}, \varepsilon \to \varepsilon$
q_2
q_3
$\mathtt{b}, \varepsilon \to \mathtt{a}$
$\varepsilon, \varepsilon \to \$$
$\mathtt{a}, \varepsilon \to \varepsilon$
$\mathtt{a}, \varepsilon \to \varepsilon$
q_0
q_4
q_5
$\mathtt{a}, \varepsilon \to \varepsilon$
$\mathtt{a}, \varepsilon \to \varepsilon$
$\mathtt{a}, \varepsilon \to \varepsilon$
$\mathtt{b}, \mathtt{a} \to \varepsilon$
q_9
q_6
q_7
$\varepsilon, \varepsilon \to \varepsilon$
$\mathtt{b}, \varepsilon \to \varepsilon$
$\mathtt{b}, \varepsilon \to \varepsilon$
q_8

Problem 2 (200 pts)**.**

Consider the following languages on $\Sigma = \{\mathtt{0}, \mathtt{1}, \mathtt{2}\}$:

- $L_2 = \{\mathtt{0}^m\mathtt{1}^n : m, n \geq 0\}$
- $L_3 = \{\mathtt{0}^m\mathtt{1}^n : n \geq m\}$
- $L_4 = \{\mathtt{1}^m\mathtt{2}^n : m, n \geq 0\}$
- $L_5 = \{\mathtt{1}^m\mathtt{2}^n : n \leq m\}$
- $L_6 = \{\mathtt{0}^\ell\mathtt{1}^m\mathtt{2}^n : \ell, m, n \geq 0\}$

Show that the following are context free languages:

- $(L_2 \cap L_3) \cup L_6$
- $(L_2 \cap L_4) \circ L_3$
- $L_3 \cup L_5$

Problem 3 (200 pts)**.**

For the following language L_7, give a PDA which accepts it:

$$L_7 = \{\texttt{0}^w\texttt{1}^x\texttt{2}^y\texttt{3}^z : w + x + y = 2z\}$$

Problem 4 (250 pts)**.**

For the purposes of this exercise, we will use input alphabet $\Sigma = \{\texttt{0,1}\}$. Using the Turing machine simulator found at https://alistat.eu/online/turingmachinesimulator, write code to simulate a Turing machine which performs the following objective:

Given two input strings $a = a_1a_2 \cdots a_k$ and $b = b_1b_2 \cdots b_k$, which are separated on the tape by a single blank space, compute $a \oplus b$. Assume the tape head starts over the leftmost symbol in a, with b to its right (again, separated by a single space). You can assume that $|a| = |b|$, and that the bits are indexed similarly (i.e. that without loss of generality the least significant bit of a is on the left boundary of a and the least significant bit of b is on the same boundary).

When your machine is done, the following should hold:

- It should end in an accept state.
- The string (bit vector) representing $a \oplus b$ should begin to the right of the initial position of the b string, separated from it by a single blank space (note that there is no requirement to maintain b, however please see the bonus below).
- The tape head should be positioned over the least significant bit of the output, $a \oplus b$.
- Only left and right transitions may be used. *You will not receive full credit if you use the 'stay' transition*, even though the website provides for it.
- **Bonus (100 pts)** retain/reinstate the strings a and b in their original position at the end of computation.

For example, on input...

```
...␣␣0010101␣1001100␣␣...
```

valid output might look like...

```
...␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣1011001␣␣...
```

Note that this simulator uses doubly infinite tape. The tape alphabet will be at a minimum, $\Gamma = \Sigma \cup \{\text{␣}\}$. You may feel free to use additional symbols in your tape alphabet, as long as they have been removed by the end of computation. Additionally, the source code for the Lecture 7 machine will be provided to you alongside this document. You may use it to get used to the syntax and might find some of its logic helpful in constructing your machine.

Submit the simulator code for your machine as a **separate** text file within your myCourses submission.