1 Modeling Concurrent Processes with LTL

- Regarding Problem 1 of problem set 7 (posted Friday):
  - Two processes \( P \) and \( Q \) running concurrently with a shared variable \( n \) storing some integer \( i \in \{-2, -1, 0, 1, 2\} \). As for the code itself, I didn’t write it down, but the basic idea is that \( n \) is initialized to 0, process \( Q \) decrements \( n \) while \( P \) increments \( n \) (each by 1), and once \( n = -2 \), process \( Q \) terminates and \( n \) is re-initialized to 0.

- To model this system, think of a model \( M \) with 8 states:
  - \( S[P, Q, i] \), with \( i \in \{-2, -1, 0, 1, 2\} \), representing when both \( P \) and \( Q \) are running.
  - \( S[P, i] \) with \( i \in \{0, 1, 2\} \) when only \( P \) is running.

- Such a model can be represented in terms of the following directed graph:
2 Regular Expressions and Regular Language

- Consider a directed graph with nodes representing states of some system, with each edge labeled either \( a \) or \( b \). Thus, any path in the system will be represented by a series of \( a \)'s and \( b \)'s, and will look something like \( aabababa \cdots \).

- The expression \( L(A) \) is used to denote a particular execution path through the system to one of its final states. More formally, \( L(A) = "The language recognized by the automaton \( A \)."

  - "+" means 'or', so \( (a + b) \) means that either path \( a \) or path \( b \) can be taken.
  - An expression like \( a^* \) means that the path \( a \) can be taken an arbitrary number of times, like when a node has an edge which leads immediately to itself.
  - So, an expression like \( (a+b)^*b(a+b) \) means "either path \( a \) or \( b \) is taken (possibly both, alternating) an arbitrary number of times, followed by path \( bb \), followed by either path \( a \) or path \( b \)."

3 \( \omega \)-Regular Expressions and \( \omega \)-Regular Language

- Extends finite automata for infinite paths.

  - One type of \( \omega \)-automata is called a "Buchi Automaton". It accepts an infinite input sequence if there exists a run of the automaton which visits at least one of the final states infinitely often.
  - Relevant for deciding validity for models in LTL.