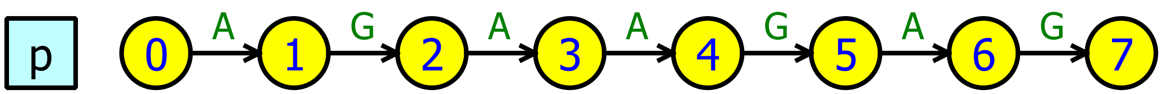
Cairo University
Faculty of Computers and Artificial Intelligence
Computer Science Department

Biological Sequence Analysis **Knuth-Morris-Pratt Algorithm** Dr. Amin Allam

1 Failure links automaton

The Knuth-Morris-Pratt (KMP) algorithm allows efficient searching for patterns in non-indexed strings. The first step in the KMP algorithm is to construct the failure links automaton for the pattern that we need to search for. The following animated figure explains how to construct the failure links automaton for the pattern `AGAAGAG`. Each mouse click on the following animation moves one step:



Node 0 does not have a failure link
because it is the first node

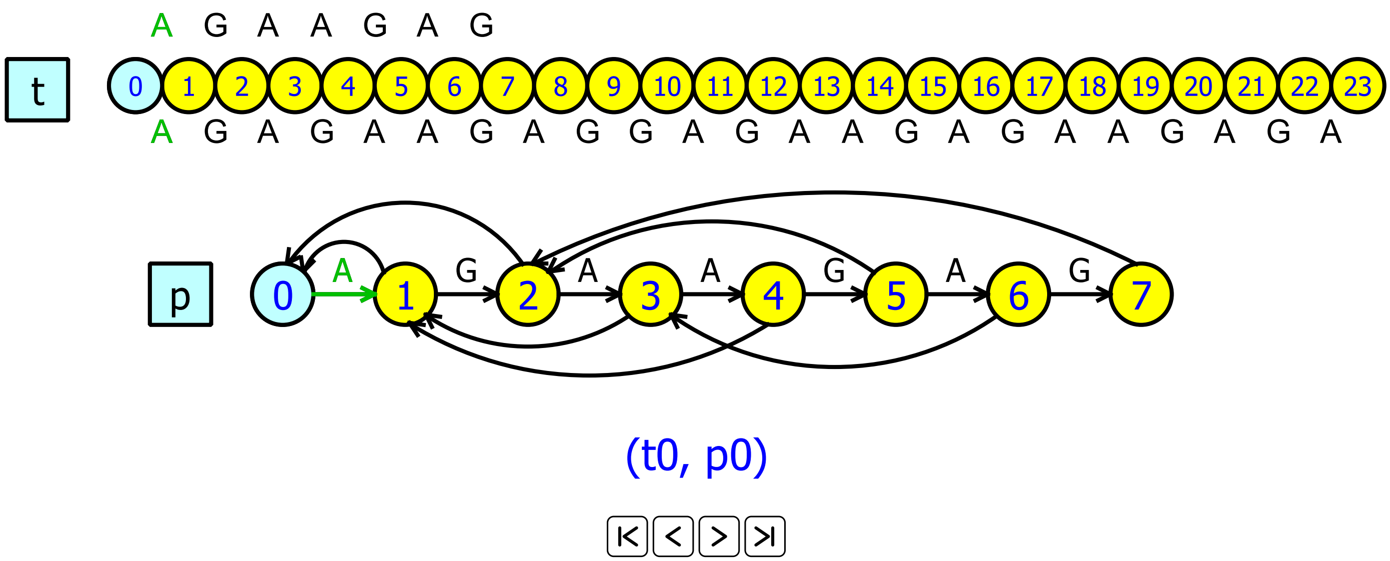
Let p be a pattern of size $|p|$ for which we need to search. To construct the failure links automaton of p, we start by constructing a sequence of $|p| + 1$ nodes $\{p_0, p_1, ..., p_{|p|}\}$ connected by labeled directed edges. For all characters in p, there exists an edge labeled with the i^{th} character of the pattern starts from the i^{th} node and ends at the $(i + 1)^{th}$ node.

Then for each node other than the first one, construct a failure link that starts from that node and goes backward to another node as follows: Consider the pattern prefix $(0..i)$ which is the sequence of characters labeling the edges from node p_0 to node p_i. The failure link starting from node p_i must end at node p_j, where j is the length of the longest proper suffix of $(0..i)$ that is also a prefix of the pattern p. The proper suffixes of a string are all suffixes of that string except the string itself. If no such suffix exist, the failure link starting from node p_i must end at node p_0.

For example, to find the destination of the failure link starting from node p_5 in the pattern `AGAAGAG`, consider the pattern prefix `AGAAG` which ends at node p_5. The proper suffixes of that prefix, ordered from longest to shortest, are: (`GAAG`, `AAG`, `AG`, `G`). The longest proper suffix of these 4 suffixes that is also a prefix of the pattern is `AG` (because the pattern starts with `AG`). Since the length of this proper suffix is 2, the failure link starting from node p_5 must end at node p_2.

2 Knuth-Morris-Pratt algorithm

The following animated figure is an example of applying the Knuth-Morris-Pratt (KMP) algorithm to search for the pattern `AGAAGAG` inside the text `AGAGAAGAGGAGAAGAGAAGAGA`. Each mouse click on the following animation moves one step:



To search for pattern p of size $|p|$ in text t of size $|t|$, the KMP algorithm proceeds as follows: Construct the failure links automaton of p. Construct an automaton for text t which is similar to the automaton of p without failure links. The algorithm keeps track of two nodes: a node in the automaton of t and another node in the automaton of p. The algorithms starts from the state (t_0, p_0) which are the initial nodes in the t and p automata. The algorithm terminates when it reaches the state $(t_{|t|}, p_0)$. The algorithm reports an occurrence of p inside t starting from $t_{i-|p|}$ whenever the algorithm reaches a state $(t_i, p_{|p|})$ for all i.

Suppose the algorithm is at state (t_i, p_j):
- If the edges $t_i \to t_{i+1}$ and $p_j \to p_{j+1}$ have equal labeling characters, go to (t_{i+1}, p_{j+1}).
- Otherwise:
  - If $j = 0$, go to (t_{i+1}, p_j).
  - Otherwise, go to $(t_i, f[p_j])$, where $f[p_j]$ is the destination node of the failure link of node p_j.

The complexity of the KMP algorithm is $O(|t| + |p|)$, which is asymptotically better than the naive brute force algorithm $O(|t| \cdot |p|)$. The failure links help to avoid the unnecessary computations done by the naive algorithm. The moving pattern p at the top of the above animation helps to understand how the algorithm avoids unnecessary computations. For official proofs regarding the correctness and complexity of the KMP algorithm, consult the reference text book.