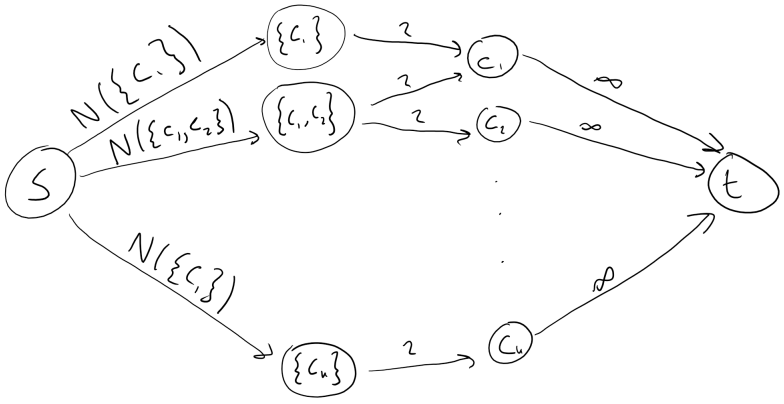
CSC 226 Problem Set 4 Written Part

Oliver Tonnesen
V00885732

November 28, 2018

1 Tragic Comedy and Network Flows

We represend this problem with the following network graph:
Let $N(S) \equiv$number of people p such that p tolerates exactly S, $S \subseteq \{c_1, c_2, \ldots, c_k\}$.



Each comedian $c_i \in \{c_1, c_2, \ldots, c_j\}$ has a corresponding vertex t_i in the network graph. Each subset $v_k \subseteq \{c_1, c_2, \ldots, c_j\}$ has a corresponding vertex n_k such that n_k flows to t_i if and only if $c_i \in V_k$. Every passenger p_l represents the one unit of flow from the source vertex to the vertex n_k such that the subset of comedians that p_l will tolerate is equal to v_k.

If we restrict the capacity of the flow from any n_k to any n_k to 2, then this network graph fully represents the given problem. If we now apply an algorithm to this graph to determine its maximum flow, we will have efficiently answered the question of whether or not it is possible to evacuate everyone onboard.

2 Max-Flow with Node Capacities

We can simply construct a standard network graph given such a node-capacited network. The Ford-Fulkerson algorithm can then be applied directly. We construct the network graph as follows:
For every vertex in the graph – starting from the sink – do the following:
Let each edge flowing to it have the nodes capacity. Let every edge flowing from it have capacity equal to min(edge's currenct capacity, node's capacity). Perform this process recursively until each edge has a capacity.

This transformation takes $O(n+m)$ operations. Note that the Ford-Fulkerson algorithm takes $O(m|f^*|)$ operations, where f^* is the maximum flow in the network. We know $m \geq n-1$, so $n+m \in O(m|f^*|)$, and our transformation does not change the asymptotic running time of the solution.

3 Min-cut and increasing all edge capacities by 1

Suppose (A^*, B^*) is no longer a minimum cut after the transformation. Then there must exist some (A', B') such that:

$$\sum_{a\in A^*}\sum_{b\in B^*} f(a,b) - \sum_{a\in A^*}\sum_{b\in B^*} f(b,a) \leq \sum_{a\in A'}\sum_{b\in B'} f(a,b) - \sum_{a\in A'}\sum_{b\in B'} f(b,a) \quad (1)$$

and

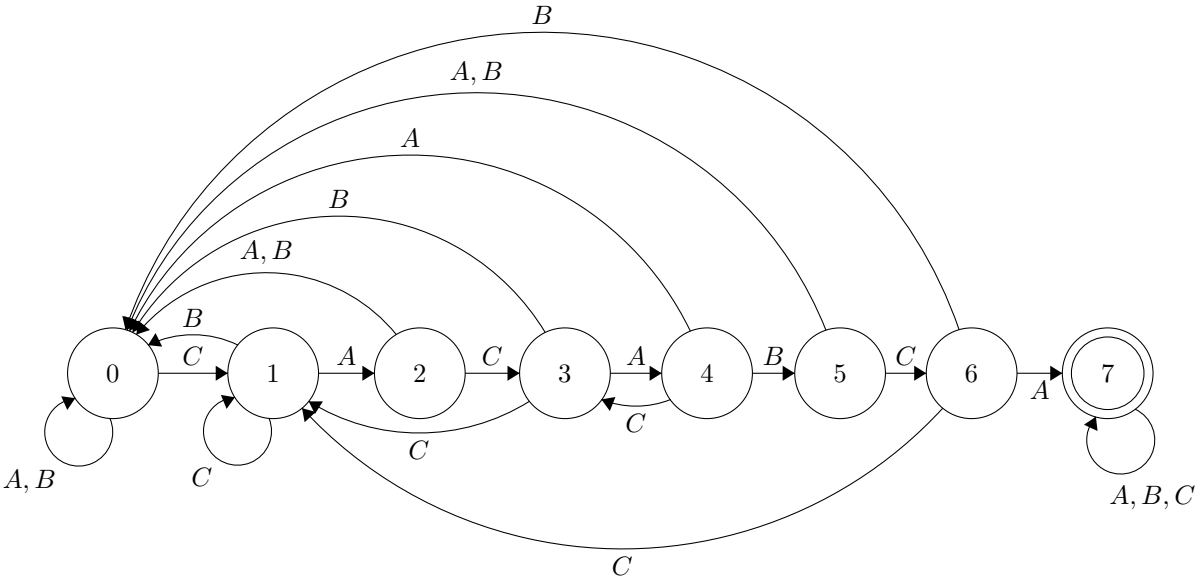
$$\sum_{a\in A^*}\sum_{b\in B^*} [f(a,b)+1] - \sum_{a\in A^*}\sum_{b\in B^*} [f(b,a)+1] > \sum_{a\in A'}\sum_{b\in B'} [f(a,b)+1] - \sum_{a\in A'}\sum_{b\in B'} [f(b,a)+1] \quad (2)$$

$$\begin{aligned}
&\sum_{a\in A^*}\sum_{b\in B^*} [f(a,b)+1] - \sum_{a\in A^*}\sum_{b\in B^*} [f(b,a)+1] \\
&= \left[\sum_{a\in A^*}\sum_{b\in B^*} f(a,b) + \sum_{a\in A^*}\sum_{b\in B^*} 1\right] - \left[\sum_{a\in A^*}\sum_{b\in B^*} f(b,a) + \sum_{a\in A^*}\sum_{b\in B^*} 1\right] \\
&= \sum_{a\in A^*}\sum_{b\in B^*} f(a,b) - \sum_{a\in A^*}\sum_{b\in B^*} f(b,a)
\end{aligned}$$

$$\begin{aligned}
&\sum_{a\in A'}\sum_{b\in B'} [f(a,b)+1] - \sum_{a\in A'}\sum_{b\in B'} [f(b,a)+1] \\
&= \left[\sum_{a\in A'}\sum_{b\in B'} f(a,b) + \sum_{a\in A'}\sum_{b\in B'} 1\right] - \left[\sum_{a\in A'}\sum_{b\in B'} f(b,a) + \sum_{a\in A'}\sum_{b\in B'} 1\right] \\
&= \sum_{a\in A'}\sum_{b\in B'} f(a,b) - \sum_{a\in A'}\sum_{b\in B'} f(b,a)
\end{aligned}$$

So clearly (1) and (2) cannot both be correct, and therefore no such cut (A', B') exists. □

4 Knuth-Morris-Pratt



	0	1	2	3	4	5	6	7
A	0	2	0	4	0	0	7	7
B	0	0	0	0	5	0	0	7
C	1	1	3	1	3	6	1	7

5 Rabin-Karp and Wildcards

Since we know at which index the wildcard occurs, we can simply modify the hash function to disregard whatever character falls on that index when computing the hash value.