CS525 Winter 2012 \ Class Assignment #7 3/14/2012

Aylin Caliskan

Nicholas Milkovits

Ariel Stolerman

7.9)

Let $TRIANGLE = \{\langle G \rangle \mid G\ is\ an\ undirected\ graph\ that\ contains\ a\ triangle\}$. Following is a proof that $TRIANGLE \in P$:

Let M be a TM defined as follows:

"For input G:

- Make sure G is a proper encoding of an undirected graph, otherwise *reject*.
- Enumerate over all possible triplets of nodes in G, and check if all are connected to each other.
- If found such triplet, *accept*. Otherwise, *reject*."

Note: we can assume a multi-tape TM for simplicity of running-time analysis (i.e. not take into consideration going back and forth over a single tape), as it maintains "polynomiality".

Correctness:

Clearly if G contains a triangle, it will be found and the machine will accept, and if not – the machine will reject.

Running time:

Denote the number of vertices n and the number of edges m, and $m = O(n^2)$.

- The first step can take $O((n+m)^2)$ time, including copying the nodes and edges to different tapes, and scanning for correctness of both lists (no duplicate nodes, edges encoded over nodes correctly etc.).
- There are $\binom{n}{3} = O(n^3)$ possible triplets. Each triplet needs to be checked for 3 adjacencies (of each possible pair), each takes $O(m)$ time for traversing over all edges. The total running time of this step is therefore $O(n^3 m)$.

The total running time is $O((n+m)^2 + n^3 m)$ which is polynomial in the length of the input $n+m$, therefore $TRIANGLE \in P$.

7.13)

Let $PP = \{\langle p, q, t \rangle \mid p = q^t\ where\ p, q\ are\ permutations\ on\ \{1, \dots, k\}\ and\ t\ is\ a\ binary\ integer\}$.

Following is a proof that $PP \in P$:

Let M be a TM defined as follows:

"On input $\langle p, q, t \rangle$:

- Let $q^1 \coloneqq q$. For $i = 2, \dots, \lceil \lg t \rceil$:
  - Calculate $q^i = q^{i-1} \circ q^{i-1}$, where $\circ$ is denoted composition
- Construct q' as the composition of all q^i for all i such that the ith bit of t is 1.
- Compare p and q'. If all transitions are equal, *accept*. Otherwise, *reject*."

Correctness:

We can calculate each q^i from the previous q^{i-1} by composing the latter with itself, as it simulates doubling the number of steps. Eventually, since t is a binary number, we can simulate composing q for t times by composing all elements in the sum of powers of 2 that form t – namely all q^i where the ith bit of t is 1. Therefore the construction of $q' = q^t$ is correct. The rest is trivial.

Running time:

The loop that calculates all q^i runs $O(\lg t)$, and since t is a binary integer it is equal to $O(n)$. Each iteration can take $O(n^2)$, as for each element of the $O(n)$ elements in the permutation q^{i-1} we need to scan $O(n)$ transitions in q^{i-1} in order to "double" the permutation. The total running time for constructing all q^i is therefore $O(n^3)$. Constructing q' takes similar time: calculating at most $O(n)$ compositions, each $O(n^2)$. The last step of comparing the permutations can be done in linear time (using multi-tape TM; $O(n^2)$ otherwise). The total running time is therefore $O(2n^3 + n^2) = O(n^3)$, an so $PP \in P$, as required.

7.14)

Following is a proof that P is closed under the star operation:

We describe a dynamic-programming algorithm. Let $A \in P$ be a polynomially-decidable language, and let M be a TM poly-time decider for A. Following is a description of a TM M^* which is a decider for A^*:

"On input w:

- Let n be the length of the input $w = w_1 w_2 \ldots w_n$.
- Initialize a table of size $n \times n$, denoted S which will hold at each cell where $i \leq j$: $S_{ij} = \begin{cases} 1, & w_i w_{i+1} \ldots w_j \in A^* \\ 0, & otherwise \end{cases}$
- For $i = 1, \ldots, n$:
  - Simulate M on w_{ii}. If it accepts, mark $S_{ij} = 1$.
  - Otherwise, mark $S_{ij} = 0$.
- For all the rest of the cells where $i \leq j$:
  - Simulate M on $w_i \ldots w_j$. If it accepts, mark $S_{ij} = 1$.
  - Otherwise, for $k = i, i+1, \ldots, j-1$:
    - If $S_{ik} = 1$ and $S_{(k+1)j} = 1$, mark $S_{ij} = 1$ and break the loop.
  - Otherwise, mark $S_{ij} = 0$
- If $S_{1n} = 1$, *accept*. Otherwise, *reject*."

Correctness:

The construction of the table is based on the fact that if $u, v \in A^*$ then $w = uv \in A^*$ as well. Each substring of w is checked for that or for membership in A (using the decider of A) – the only two possibilities in which $w \in A^*$. Therefore eventually S_{1n} will be 1 iff $w_1 \ldots w_n = w \in A^*$.

Running time: The construction is clearly polynomial, therefore P is closed under the star operation.