3P17/4P17 COMPUTABILITY 2014-2015 Lecture Handout 13

The class P

P is the class of languages that are decidable in polynomial time by a deterministic single-tape Turing machine. i.e.,

$$\mathbf{P} = \bigcup_{k=1}^{\infty} \mathbf{TIME}(n^k)$$

Theorem 1. *Let*

$$PATH = \{\langle G, s, t\rangle : \ G \textit{ is a directed graph that has a directed path from } s \textit{ to } t\}.$$

Then $PATH \in \boldsymbol{P}$.

Proof. A polynomial time algorithm M for PATH is as follows:

$\underline{M}$
On input $\langle G, s, t\rangle$ where G is a directed graph with nodes s and t:

(1) Place a mark on node s.

(2) Look for a directed edge (a, b) from a to b where a is marked and b is not marked. If no such edge can be found, go to (4), otherwise go to (3).

(3) Mark node b, go to (2).

(4) If t is marked, then accept, otherwise, reject.

Lines (1) and (4) are each executed once. They can each be done in polynomial time since they just involve searching through the representation and recording that a node has been marked.

Lines (2) and (3) are executed at most n times, where n is the number of nodes in the graph. This is because a node is marked at most once, and if a node is not marked in (3) then the loop breaks and it has gone to (4).

Line (2) can be executed in polynomial time because it involves a scan of the input, and line (3) just involves recording a marking, which can also be done in polynomial time.

Hence, all lines can be executed in polynomial time and have a polynomial number of executions. Hence, M runs in polynomial time.

$\square$