

## Problem 8.3

### Reachable

Require: a DAG  $(V, E)$  and a source node  $s$

Ensure: a *reachable* array of booleans that stores whether or not each  $v \in V$  is reachable from  $s$ .

```
function REACHABLE( $V, E, s$ )
  reachable  $\leftarrow$  new array[size of  $V$ ]
  for  $v \in [V^*(0), V(s))$  do
    reachable[ $v$ ]  $\leftarrow$  false
  end for
  reachable[ $s$ ]  $\leftarrow$  true
  for  $v \in (V^*(s), V(\text{size of } V - 1))$  do
    for  $\{u \mid (u, v) \in E\}$  do
      reachable[ $v$ ]  $\leftarrow$  reachable[parent[ $v$ ]]
    end for
  end for
end function
```

### Number of Paths

$NP_s$  denotes a function that gives the longest paths from a source node  $s$ . The summation in o/w case is the sum of the function NP applied to all the parent nodes of the inputted node. This is all the nodes  $u$  such that there as an edge starting at  $u$  and ending at  $v$ , the argument passed in.

$$NP_s(v) = \begin{cases} 1 & \text{if } v = s \\ 0 & \text{if } \neg \text{reachable}_s(v) \\ \sum_{u \mid (u,v) \in E} NP_s(u) & \text{o/w} \end{cases}$$