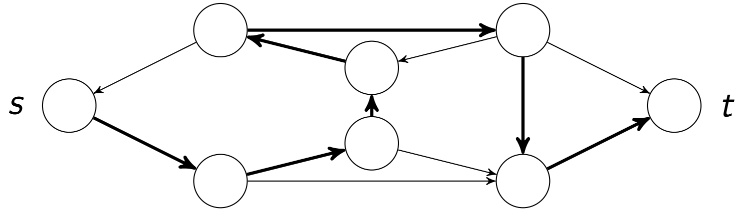
Hamiltonian Path I

- For some problems it is difficult to find an algorithm in P
- We first discuss an example of finding a Hamiltonian Path
- Definition: for a given directed graph find a path going through all nodes once
- Fig 7.17

Hamiltonian Path II



- HAMPATH $= \{\langle G, s, t\rangle \mid G$: directed, a Hamiltonian path from s to $t\}$
- A brute-force way: checking all possible paths

  But the number is exponential
- Polynomial verification

  for a path, in P time $\Rightarrow$ a Hamiltonian path or not

Hamiltonian Path III

- This is an example where verification is easier than determination

Compositeness I

- We discuss another example where verification is easier than determination
- An integer is composite if

$$x = pq, p > 1, q > 1$$

- Given x, difficult to find p, q
- Given x, p, q easily verify $x = pq$ or not

Not polynomial verifiable I

- Some problems are difficult so even a polynomial verifier cannot be easily obtained
- $\overline{HAMPATH}$: given $\langle G, s, t \rangle$ no Hamiltonian path from s to t
- Verification may still be difficult
- Given s and t it seems we still need to check all paths

Verifier I

- Definition: an algorithm V is a verifier of a language A if

$$A = \{w \mid V \text{ accepts } \langle w, c \rangle \text{ for some strings } c\}$$

- Example: compositeness. V accepts $\langle w, c \rangle = \langle x, p \rangle$, where p is a divisor
- Example: Hamiltonian path. V accepts

$$\langle w, c \rangle = \langle \langle G, s, t \rangle, \text{a path from } s \text{ to } t \rangle$$

- c is called a "certificate"

Verifier II

- Definition: a polynomial verifier if it takes polynomial time of $|w|$
- A: polynomially verifiable if $\exists$ a polynomial verifier
- Note that we measure time on $|w|$ without considering $|c|$
- For a polynomial verifier, $|c|$ should be in polynomial of $|w|$

  Otherwise, reading $|c|$ already non-polynomial

NP I

- NP is a class of languages
- Definition: a language $\in$ NP if it has a polynomial verifier
- We will prove that this definition is equivalent to that the language is decided by nondeterministic polynomial TM
- This is where the name comes from
- Some use this as the definition
- Note that for nondeterministic TM the running time is by checking the longest branch

NP II

- Definition:

$$\begin{aligned} & \mathrm{NTIME}(t(n)) \\ = & \{L \mid L \text{ decided by } O(t(n)) \text{ nondeterministic TM}\} \end{aligned}$$

- $\mathrm{NP} = \cup_k \mathrm{NTIME}(n^k)$

NTM for HAMPATH I

- A list $p_1 \cdots p_m$ is nondeterministically chosen
- For each list:
  1. Check repetitions
  2. Check $s = p_1$; $t = p_m$
  3. Check that for $i = 1 \ldots m-1, (p_i, p_{i+1})$ is an edge of G
- Cost on each list is polynomial:

  repetitions: $O(m^2)$

  $s = p_1, t = p_m : O(m)$

  edge check: $O(m^2)$