# The Class NP

# Another Example of Problems in P

- Theorem: Every CFL is a member of P.

  Proof. Give a polynomial time algorithm for each CFL

  - Let L be a CFL generated by CFG G in CNF.
  - Any derivation of a string $w$ has $2|w| \Leftrightarrow 1$ steps.
    - brute-force search: $2^{O(|w|)}$
  - Use dynamic programming technique: solve smaller problems first.
    - e.g. Fibonacci function:
      $f(n) = f(n \Leftrightarrow 1) + f(n \Leftrightarrow 2)$, and $f(0) = f(1) = 1$.
  - For an input string $w = w_1 w_2 \cdots w_n$, $S \Rightarrow^* w$?
    - subproblems: determine the variables in G that generates each substrings of $w$.
      From small to large: substrings of length 1, 2, 3, $\cdots$
    - Use an $n \times n$ table

# The Class NP

- TIME$(t(n)) = \{L \mid L$ is decided by a $O(t(n))$ time deterministic TM $\}$
- P $= \cup_{k \geq 0}$ TIME$(n^k)$
  "tractable"
- NTIME$(t(n)) = \{L \mid L$ is decided by a $O(t(n))$ time nondeterministic TM $\}$
- NP $= \cup_{k \geq 0}$ NTIME$(n^k)$
- Another definition: NP $= \{L \mid L$ can be verified in polynomial time $\}$

# Problems in NP

- Known to be in NP, but not known if in P:
  - Traveling salesman (TSP)
  - Hamiltonian circuit (HC)
  - Satisfiability (SAT)
  - Graph coloring (GC)
  - Vertex cover (VC)
  - Independent set (IS)
  - Clique
  - Knapsack, etc.
- Decision problems (languages) derived from search (optimization) problems.

# Traveling Salesman Problem (TSP)

- TSP: Given a complete graph with weights on edges, find a cycle of least total weight that visits each vertex exactly once.
- Decision problem: TSP $= \{< G, k > \mid G$ is a complete graph with weights on edges that contains a cycle of total weight $\leq k$ that visits each vertex exactly once. $\}$
- Relation between TSP optimization and decision problems.
  - solve optimization problem efficiently $\Leftrightarrow$ solve decisoin problem efficiently

# TSP in NP

- Decision problem: TSP $= \{< G, k > \ | G$ is a complete graph with weights on edges that contains a cycle of total weight $\leq k$ that visits each vertex exactly once. $\}$
- Proof:
  - Given a graph G, integer $k$, see if $< G, k > \in$ TSP
  - Nondeterministically write down a sequence of vertices
  - Check all of the following deterministically in polynomial time
    - the sequence is a simple cycle
    - it visits each vertex exactly once
    - the total cost $\leq k$

# Proof of A Language in NP

- Proof:
  - Nondeterministically "guess" the solution
  - Verify deterministically in polynomial time that the solution is correct.
- Examples
  - HC: Given a graph G, $\exists$ a circuit visiting each vertex exactly once?
    $= \{ < G > \mid G$ is a graph containing a circuit that $\cdots \}$
  - SAT: Given a Boolean formula in conjunctive normal form (C.N.F.), $\exists$ a satisfying assignment?
    $= \{ B \mid B$ is a Boolean formula in C.N.F. that is satisfiable by some truth assignment to its variables $\}$
  - CLIQUE: $\{ < G = (V, E), k > \mid \exists$ subset $V'$ of $V$ of size $\geq k$ and $\forall u, v \in V'$, $(u, v) \in E \}$

# P versus NP

- P : problems can be solved (find solution) in polynomial time.

  NP: problems whose solutions, if given, can be verified in polynomial time.
- A great problem: $P = NP?$