

# Logical Characterization of P

**Definition 1.** A graph-theoretic property  $\mathcal{G}$  is expressible in **Horn existential second order logic with successor** if there is a Horn existential second-order expression  $\phi$  with two binary relational symbols  $G$  and  $S$ , such that the following is true:

For any model  $M$  appropriate for  $\phi$  such that  $S^M$  is a linear order on the nodes of  $G^M$ ,  $M \models \phi$  iff  $G^M \in \mathcal{G}$ .

**Theorem 1.** The class of all graph-theoretic properties expressible in Horn existential second-order logic with successor is precisely  $P$ .

# Proof

We showed one direction earlier.

Given a deterministic TM  $M$  deciding graph-theoretic property  $\mathcal{G}$  within time  $n^k$ , we shall construct an expression in Horn existential second-order logic (with successor  $S$ ) that expresses  $\mathcal{G}$ .

The construction is identical to the previous proof.

There is no  $C_0$  or  $C_1$  and the expression is Horn.

# An Alternate Characterization of NP

**Definition 2.** Let  $R \subseteq \Sigma^* \times \Sigma^*$  be a binary relation on strings.

(1)  $R$  is **polynomially balanced** if  $(x, y) \in R$  implies  $|y| \leq |x|^k$  for some  $k \geq 1$ .

(2)  $R$  is **polynomially decidable** if there is a deterministic TM deciding the language  $\{x; y : (x, y) \in R\}$  in polynomial time.

**Theorem 2.** Let  $L \subseteq \Sigma^*$  be a language.  $L \in NP$  iff there is a polynomially decidable and polynomially balanced relation  $R$  such that  $L = \{x : (x, y) \in R \text{ for some } y\}$ .

## Proof

Suppose such an  $R$  exists.

Then  $L$  is decided by the following NTM  $M$ :

On input  $x$ ,  $M$  guesses a  $y$  of length at most  $|x|^k$ .

Then uses the polynomial algorithm on  $x; y$  to test whether  $(x, y) \in R$ .

If *yes* then  $M$  accepts else rejects.

Suppose  $L \in NP$ . Let NTM  $N$  decide  $L$  in time  $|x|^k$ .

Define  $R$  as:  $(x, y) \in R$  iff  $y$  is the encoding of an accepting computation of  $N$  on input  $x$ .

$R$  is polynomially balanced and polynomially decidable.

Since  $N$  decides  $L$ ,  $L = \{x : (x, y) \in R \text{ for some } y\}$ .

# NP has Short Certificates

A **certificate** is a “proof” that an instance of a decision problem has a “yes” answer.

A verification algorithm takes an instance of the problem and a **certificate** and uses it to verify whether the decision problem has a “yes” solution.

**NP** is the set of decision problems that have polynomial-time verifiers.

This means that for problems in NP certificates are polynomial in the size of the input and the verification algorithm takes polynomial time.

Thus for every “yes” instance of an NP problem, there is a polynomial-size certificate.

Example: HAMILTONIAN PATH  $\in$  NP.

The certificate is a permutation of the vertices and can be verified in polynomial time.

“Guess and Verify” method.