

# Logical Characterization of NP

Let  $\mathcal{G}$  be a set of finite graphs — i.e., a graph theoretic property.

We say that  $\mathcal{G}$  is expressible in existential second-order logic if there is an existential second-order logic sentence  $\exists P\phi$  such that  $G \models \exists P\phi$  iff  $G \in \mathcal{G}$ .

The computational problem corresponding to  $\mathcal{G}$  is to decide, given a graph  $G$ , whether  $G \in \mathcal{G}$ .

We can denote by NP the sets of graph-theoretic properties whose corresponding computational problem is in NP.

**Theorem 1. [Fagin's Theorem]** *The class of all graph-theoretic properties expressible in existential second-order logic is precisely NP.*

# Proof

If  $\mathcal{G}$  is expressible in existential second-order logic, we have shown that it is in NP.

We will show the converse. Let  $\mathcal{G}$  be a property in NP.

That is, there is a NTM  $M$  whether  $G \in \mathcal{G}$  for some  $G$  with  $n$  nodes in time  $n^k (k > 2)$ .

We will construct a second-order expression  $\exists P\phi$  such that  $G \models \exists P\phi$  iff  $G \in \mathcal{G}$ .

The input of  $M$  is the adjacency matrix of the graph and is given in the input string as follows: between any two entries we have  $n^{k-2} - 1$  blanks.

$P$  will be a Cartesian product of a list of new relations  $P_1, \dots, P_m$ .

That is, we will describe an equivalent relation of the form  $\exists P_1 \dots \exists P_m \phi$ .

## Relation $P$

The  $P_i$ 's are:

(1)  $S$  is a binary relation symbol which represents a successor function over nodes of  $G$ .

That is, in any model  $M$  of  $\phi$ ,  $S$  will be a relation isomorphic to  $\{(0, 1), (1, 2), \dots, (n - 2, n - 1)\}$ .

$S$  allows us to identify nodes of  $G$ , e.g.,

$v_0(x) = \forall y \neg S(y, x)$  identifies  $x = 0$ .

$v_{n-1}(x) = \forall y \neg S(x, y)$  identifies  $x = n - 1$ .

## Relation $P$

We can also define a first-order expression  $S_k(X, Y)$  with  $2k$  free variables which captures the successor function of  $k$  tuple  $X = (x_1, \dots, x_k)$ :

$S_k(X, Y)$  iff  $Y$  encodes the  $k$ -digit  $n$ -ary number that comes after the one in encoded by  $X$ .

That is,  $S_k$  is a successor function in  $\{0, 1, \dots, n^k - 1\}$ .

$S_j$  can be defined inductively as follows:

$$S_1 = S.$$

$$S_j = \forall(\text{variables})[S(x_j, y_j) \wedge (x_1 = y_1) \wedge \dots \wedge (x_{j-1} = y_{j-1})] \vee [v_{n-1}(x_j) \wedge v_0(y_j) \wedge S_{j-1}(x_1, \dots, x_{j-1}, y_1, \dots, y_{j-1})]$$

Since we can “count”  $k$ -tuples of variables,  $S_k$  can be used to represent numbers between 0 and  $n^k - 1$ .

## Relation $P$

(2) A  $2k$ -ary relation symbol  $T_\sigma$  for each symbol  $\sigma$  in the computation table.

$T_\sigma(X, Y)$  holds if the  $(i, j)$ th entry is  $\sigma$ , where  $X$  encodes  $i$  and  $Y$  encodes  $j$ .

(3) We have two  $k$ -ary relation symbols  $C_0$  and  $C_1$  corresponding to the two nondeterministic choices.

$C_0(X)$  holds if at the  $i$ th step ( $X$  encodes  $i$ ), the choice 0 is made.

Similarly  $C_1(X)$ .

## Expression $\phi$

(1)  $S$  is a successor relation.

(2) The top row and extreme columns of  $T$  should be legal:

If  $X$  encodes 0 then  $T_{\sqcup}(X, Y)$  unless the last  $k - 2$  components of  $y$  are all 0.

In that case, it is  $T_1(X, Y)$  or  $T_0(X, Y)$  depending on the adjacency matrix of  $G$ .

If  $Y$  encodes 0 then  $T_{\triangleright}(X, Y)$  and if  $Y$  encodes  $n^k - 1$  then  $T_{\sqcup}(X, Y)$ .

## Expression $\phi$

(3) All remaining entries of  $T$  are filled according to the transition relation of  $M$ .

We can express the transition relation of  $M$  as a set of quintuples  $(\alpha, \beta, \gamma, c, \sigma)$  where  $c \in (0, 1)$  is a nondeterministic choice and the others are table symbols.

Each quintuple means that whenever  $T(i - 1, j - 1) = \alpha$ ,  $T(i - 1, j) = \beta$ , and  $T(i - 1, j + 1) = \gamma$  and choice  $c$  was made at the  $(i - 1)$ th step, then  $T(i, j) = \sigma$ . We express this in  $\phi$  as:

$$[S_k(X', X) \wedge S_k(Y', Y) \wedge S_k(Y, Y'') \wedge T_\alpha(X', Y') \wedge T_\beta(X', Y) \wedge T_\gamma(X', Y'') \wedge C_c(X')] \Rightarrow T_\sigma(X, Y)$$

## Expression $\phi$

(4) One nondeterministic choice is taken at each step.

$$(C_0(X) \vee C_1(X)) \wedge (\neg C_0(X) \vee \neg C_1(X)).$$

(5) The machine ends accepting.

$$\Theta(X, Y) \Rightarrow \neg T_{no}(X, Y) \text{ where}$$

$\Theta(X, Y)$  states that  $X$  encodes  $n^k - 1$  and  $Y$  encodes 1.

We take the conjunction of all the clauses above preceded by  $5k$  universal quantifiers corresponding to the variable groups  $X, X', Y, Y', Y''$ .

$G$  satisfies the second order expression iff  $G \in \mathcal{G}$ .