Second-order logic consists of first-order logic plus the power to quantify over relations on the universe. We prove Fagin's theorem which says that the queries computable in NP are exactly the second-order existential queries. A corollary due to Stockmeyer says that the second-order queries are exactly those computable in the polynomial-time hierarchy.

7.1 Second-Order Logic

Second-order logic consists of first-order logic plus new relation variables over which we may quantify. For example, the formula \((\forall A^r)\varphi\) means that for all choices of \(r\)-ary relation \(A\), \(\varphi\) holds. Let \(\text{SO}\) be the set of second-order expressible boolean queries.

Any second-order formula may be transformed into an equivalent formula with all second-order quantifiers in front. If all these second-order quantifiers are existential, then we have a second-order existential formula. Let \((\text{SO}_\exists)\) be the set of second-order existential boolean queries. Consider the following example, in which \(R\), \(Y\), and \(B\) are unary relation variables. To indicate their arity, we place exponents on relation variables where they are quantified.

\[
\Phi_{3\text{-color}} \equiv (\exists R^1)(\exists Y^1)(\exists B^1)(\forall x) \left[ (R(x) \lor Y(x) \lor B(x)) \land (\forall y) \left( E(x, y) \rightarrow \neg(R(x) \land R(y)) \land \neg(Y(x) \land Y(y)) \land \neg(B(x) \land B(y)) \right) \right]
\]
Observe that a graph $G$ satisfies $\Phi_{3\text{-color}}$ iff $G$ is 3-colorable. Three colorability of graphs is an NP complete problem (3-COLOR). In Section 7.2, we see that three colorability remains complete via first-order reductions. It will then follow that every query computable in NP is describable in $\text{SO}_3$.

Second-order logic is extremely expressive. For this reason, it is very easy to write second-order specifications of queries. For the same reason, such specifications are not feasible to execute without further refinement (cf. Section 9.6). Recall that the first-order queries are those that can be computed on a CRAM in constant time, using polynomially many processors (Theorem 5.2). We will see that the second-order queries are those that can be computed in constant parallel time, but using exponentially many processors (Corollary 7.27).

Here are a few other examples of $\text{SO}_3$ queries.

**Example 7.1** SAT is the set of boolean formulas in conjunctive normal form (CNF) that admit a satisfying assignment (Example 2.18).

The boolean query SAT is expressible in $\text{SO}_3$ as follows:

$$\Phi_{\text{SAT}} \equiv (\exists S)(\forall x)(\forall y)((P(x, y) \land S(y)) \lor (N(x, y) \land \neg S(y))) .$$

$\Phi_{\text{SAT}}$ asserts that there exists a set $S$ of variables — the variables that should be assigned true — that is a satisfying assignment of the input formula.

**Example 7.2** Boolean query CLIQUE is the set of pairs $(G, k)$ such that graph $G$ has a complete subgraph of size $k$ (Example 2.10). The vocabulary for CLIQUE is $\tau_{k, k} = (E^2, k)$.

The $\text{SO}_3$ sentence $\Phi_{\text{CLIQUE}}$ says that there is a numbering of the vertices such that those vertices numbered less than $k$ form a clique. In order to describe this numbering it is convenient to existentially quantify a function $f$. This can be replaced by a binary relation in the usual way (Exercise 7.3). Let $\text{Inj}(f)$ mean that $f$ is an injective function,

$$\text{Inj}(f) \equiv (\forall xy)(f(x) = f(y) \rightarrow x = y)$$

$$\Phi_{\text{CLIQUE}} \equiv (\exists f^1.\text{Inj}(f))(\forall xy)((x \neq y \land f(x) < k \land f(y) < k) \rightarrow E(x, y))$$

**Exercise 7.3** Show how formula $\Phi_{\text{CLIQUE}}$ may be rewritten using an existentially quantified relation $F$ of arity two, rather than function $f$.

**Exercise 7.4** Hamiltonian-Circuit (HC) is the boolean query that is true of an undirected graph iff it has a Hamiltonian circuit, i.e., a path that starts and ends at the same vertex and visits every other vertex exactly once. Write an $\text{SO}_3$ sentence that expresses HC. [Hint: say that there exists a total ordering of the vertices that determines a Hamiltonian circuit.]