LOGICAL CHARACTERIZATIONS OF COMPLEXITY CLASSES

V.G. Naidenko

Institute of Mathematics, National Academy of Belarus,
11 Surganov str., 220072 Minsk, Belarus

Since 1974, descriptive complexity characterizes computational complexity in terms of logical languages. Fagin [1] first shown that the complexity class NP coincides with the set of problems expressible in second order existential (SO∃) logic. Stockmeyer [2] extended Fagin’s result to the polynomial-time hierarchy (PH) characterized by second order logic. Further research revealed logical characterizations for various complexity classes [3].

However, there are complexity classes such as PSPACE-complete problems, NP-complete problems, coNP-complete problems, P-complete problems, NL-complete problems, and NP ∩ coNP for which no logics were known till now. The purpose of our research is to develop logics for these classes.

Let us proceed to our results. First of all, note that it is very unlikely that one could construct a complete problem (for any reasonable complexity class) using structures which interpret only constant symbols and without function symbols. So, a vocabulary is a finite set \( \{ a_i \mid 1 \leq i \leq m \} \) of relation symbols of specified arities. A structure is a tuple \( A = ( |A|, R_1^A, \ldots, R_m^A ) \), where \( |A| \) is a nonempty finite set, and each \( R_i^A \) is a relation on \( A \) such that arity(\( R_i^A \)) = \( a_i \), \( 1 \leq i \leq m \). By a model class we mean a set structures of a fixed vocabulary \( \tau \) that is closed under isomorphism. By \( \text{STRUC}[\tau] \) we denote the model class of all structures for the vocabulary \( \tau \).

We define a logic \( \mathcal{L} \) as follows. For every vocabulary \( \tau \), the language \( \mathcal{L}(\tau) \) is the recursive set of all well-formed sentences (whose elements are called \( \mathcal{L} \)-sentences) with the symbols predefined for the logic \( \mathcal{L} \). In addition, \( \models \) is a binary relation between \( \mathcal{L} \)-sentences and structures, so that for each \( \mathcal{L} \)-sentence \( \Gamma \) with the vocabulary \( \tau \), the set \( \{ A \in \text{STRUC}[\tau] \mid A \models \Gamma \} \) denoted by \( \text{MOD}[\Gamma] \) is a model class. Also, we say that a \( \mathcal{L} \)-sentence \( \Gamma \) defines a model class \( K \) if \( K = \text{MOD}[\Gamma] \).

We will characterize a model class as a complexity theoretic problem. Let \( \mathcal{L} \) be a logic, \( \mathcal{C} \) a complexity class, and \( \tau \) a vocabulary. We say that \( \mathcal{L} \) captures \( \mathcal{C} \) if for every vocabulary \( \tau \), the following two conditions are satisfied:

1) For every \( \mathcal{L} \)-sentence \( \Gamma \) with the vocabulary \( \tau \), the model class \( \text{MOD}[\Gamma] \) belongs to \( \mathcal{C} \).

2) For every model class \( K \subseteq \text{STRUC}[\tau] \) in \( \mathcal{C} \), there exists a \( \mathcal{L} \)-sentence \( \Gamma \) that defines \( K \).

Let us proceed to our results. First of all, note that it is very unlikely that one could construct a complete problem (for any reasonable complexity class) using structures which interpret only unary relation symbols. The argument is essentially that such classes of structures are interpretable with sparse languages for which it is highly improbable to find a complete problem. Therefore, we consider complete problems on structures containing at least one binary relation in what follows.

Let \( \mathcal{C} \) denote one of the following complexity classes: NL, P, NP, coNP, and PSPACE if we allow linear order < in structures and without linear order < just the last three of them. The technique used in all cases is the same. We start out with a logic \( \mathcal{L} \) that captures the complexity class \( \mathcal{C} \) (for definiteness, by \( \mathcal{L} \) we mean one of the following logics: FO(TC), FO(LFP), SO∃, SO∀, and SO(PFP), respectively). Then, for each \( \mathcal{L} \)-sentence \( \Gamma \) and for each Turing machine \( T \), we take the sentence

\[
( \gamma \land \Gamma ) \lor ( \neg \gamma \land \Upsilon )
\]

where \( \Upsilon \) is a fixed \( \mathcal{L} \)-sentence defining some \( \mathcal{C} \)-complete problem, and \( \gamma \) is constructed so that \( \gamma \) is satisfied for a structure \( A \) if and only if on all sufficiently small structures \( B \) (taking \( \| B \| \leq \log \log \log |A| \) enough), the machine \( T \) witnesses that the models of \( \Upsilon \) are reducible to those of \( \Gamma \). It then follows that the sentence (1) defines a class which is the same as \( \Gamma \) if \( \Gamma \) defines a
$\mathcal{C}$-complete problem and is finitely different from $\mathcal{Y}$ otherwise. In the presence of linear order $<$, $\gamma$ can be chosen to be a first-order sentence, while without linear order $<$ it can be chosen to be an existential sentence. Thus, there exist logics capturing complete problems in the complexity classes NL, P, coNP, NP, and PSPACE, based on the canonical form (1).

Besides, we extend our approach beyond complete problems. One can build a class of logical sentences that defines exactly the problems being in $NP \cap coNP$. The technique is analogous to the one above. For a pair $(\Lambda, \Gamma)$ of sentences ($\Lambda$ is universal second-order and $\Gamma$ existential second-order), we take the existential second-order sentence

$$\gamma \land \Gamma$$

(2)

where $\gamma$ is an existential second-order sentence constructed so that $\gamma$ is satisfied for a structure $A$ if and only if $\Lambda$ and $\Gamma$ are equivalent for all sufficiently small structures $B$ (taking $\|B\| \leq \log \log \|A\|$ enough). Then, either $\gamma$ is identically true, or $\gamma$ defines a finite set. Therefore, MOD[$\gamma \land \Gamma$] is in $NP \cap coNP$. Thus, there exists a logic capturing $NP \cap coNP$, based on the canonical form (2).

In conclusion, we have modified a fragment of Immerman’s diagram [3] in respect to the complexity classes from NL to PSPACE, as shown in Figure 1.

Fig. 1: The World of Computability and Complexity from NL to PSPACE
(a fragment of Immerman’s diagram [3])

For purposes of clarity, in the diagram we have permitted ourself to shade areas depicting the following complexity classes: PSPACE-complete problems, NP-complete problems, coNP-complete problems, P-complete problems, NL-complete problems, and NP $\cap$ coNP for which we have developed logics for the first time.

References