DECOMPOSITIONS OF NONDETERMINISTIC
REDUCTIONS
Extended Abstract by
Klaus-Jörn Lange, Fachbereich Informatik der Universität
Hamburg, Rothenbaumchaussee 67/9, D-2000 Hamburg 13

Abstract

Nondeterministic reductions with a polynomial time bound or logarithmic space bound are characterized in terms of formal language operations like nonerasing homomorphisms and Kleene's star by relativizing the well-known equations $NP = \text{LOG}(H(DSPACE(log n)))$, $NSPACE(log n) = \text{LOG}(H(1-DSPACE(log n)))$, and $NSPACE(log n) = \text{LOG}(DSPACE(log n)^\ast)$. As corollaries we get $\Sigma^P_{k+1} = \text{LOG}(H(\Pi^P_k))$ and $O^L_{k+1} = \text{LOG}(O^L_k)^\ast$. Further on, we derive the relation $NPOL(A) = NLOG(NLOG(A))$ for every language $A$. Finally, we get that $\Delta^P_{log n}$ contains the logarithmic oracle hierarchy.

1 Introduction

This article contains parts of the author's paper [7], where the following, roughly sketched results can be found in details, in particular Theorem 3.2, Theorem 5.2, and Corollary 5.5.

In [5] several types of deterministic and nondeterministic polynomial time reducibilities were introduced, among them Turing, conjunctive Turing, and many-one reducibilities. We are now going to relate and compare these three types with respect to determinism vs. nondeterminism and polynomial time vs. logarithmic space.

In this context the problem of space bounded relativization occurs. Essentially there are three possibilities to bound the space of an oracle machine (resp. Turing transducer). One is to regard the oracle tape (resp. output tape) as a working tape. But this is not useful for sublinear, in particular logarithmic, space bounds. For instance, this case does not cover the many-one log space reduction (see [15]). A second approach, by Ladner and Lynch in [4], bounds the running time and thus the length of the oracle tape (resp. output tape) exponentially by the space bound. But this restriction seems to be too weak in the nondeterministic case, where inclusions like $NSPACE(log n) \subseteq \mathbb{P}$ or $NSPACE(log n) \subseteq DSPACE(log^2 n)$ do not relativize in this way. This led
to the third approach by Ruzzo, Simon, and Tompa in [13]. They restrict the machines to work deterministically while writing on the oracle tape (resp. output tape). After having finished an oracle query the oracle machines may continue nondeterministically.

Of course, the last two approaches coincide in the deterministic case.

In the following we will relativize the equations

\[ (1) \quad \text{NP} = \text{LOG}(H(\text{DSPACE}(\log n))), \]
\[ (2) \quad \text{NSPACE}(\log n) = \text{LOG}(H(1-\text{DSPACE}(\log n))), \text{ and} \]
\[ (3) \quad \text{NSPACE}(\log n) = \text{LOG}(\text{DSPACE}(\log n)^*), \]

where \( H(A) \) denotes the class of all nonerasing homomorphic images of elements in a language class \( A \) and \( 1-\text{DSPACE}(\log n) \) is the class of all languages, recognizable by deterministic log space bounded Turing machines with a one-way input tape.

In section 3 we will see, that equation (1) relativizes by a slight extension of the method of Cook in [2]. In section 4 equation (2) will be relativized with respect to Ladner & Lynch reducibilities, while in section 5 the relativization of equation (3) will use the Ruzzo, Simon, & Tompa reducibility.

2 Preliminaries

Let \( \lambda \) denote the empty word, \( \|v\| \) the length of a word \( v \), and \( \hat{v} \) its reversal.

For a language \( A \) let Co-\( A \) be its complement. (We do not fix the underlying alphabet, since its exchange needs intersections and unions with regular sets, only. But these Operations have no influence in the complexity of a problem.) If \( A \) is a language family, let \( H(A) \) (resp. \( A^* \), Co-\( A \)) be the class of all nonerasing homomorphic images of elements of \( A \) (resp. of their Kleene closure, of their complement).

Let \( \mathcal{CF} \) denote the class of context-free languages. Further on, set \( \mathcal{L} := \text{DSPACE}(\log n) \), \( \mathcal{NL} := \text{NSPACE}(\log n) \) and let \( 1-\mathcal{L} \) and \( 1-\mathcal{NL} \) be the corresponding classes, where the input is given one-way.

The reader is assumed to be familiar with Turing reducibilities (performed by oracle Turing machines) and many-one reducibilities (performed by Turing transducers). If we consider deterministic reducibilities under a logarithmic space bound, we get the classes \( \text{LOG}(A) \) and \( \mathcal{L}(A) \) of all sets many-one resp. Turing reducible to an (oracle) set \( A \). If the underlying oracle machine works conjunctively (see [5]; i.e. if it has to reject whenever an oracle query is answered