Machines that Can Output Empty Words

Christian Glaßer and Stephen Travers

Theoretische Informatik
Julius-Maximilians Universität Würzburg, Germany
{glasser, travers}@informatik.uni-wuerzburg.de

Abstract. We propose the e-model for leaf languages which generalizes the known balanced and unbalanced concepts. Inspired by the neutral behavior of rejecting paths of NP machines, we allow transducers to output empty words. The paper explains several advantages of the new model. A central aspect is that it allows us to prove strong gap theorems: For any class \( C \) that is definable in the e-model, either \( \text{coUP} \subseteq \mathcal{C} \) or \( \mathcal{C} \subseteq \text{NP} \). For the existing models, gap theorems, where they exist at all, only identify gaps for the definability by regular languages. We prove gaps for the general case, i.e., for the definability by arbitrary languages. We obtain such general gaps for NP, coNP, 1NP, and co1NP. For the regular case we prove further gap theorems for \( \Sigma^p_2 \), \( \Pi^p_2 \), and \( \Delta^p_2 \). These are the first gap theorems for \( \Delta^p_2 \).

1 Introduction

Bovet, Crescenzi, and Silvestri [5] and Vereshchagin [20] independently introduced leaf languages. This concept allows a uniform definition of many interesting complexity classes like NP and PSPACE. The advantage of such an approach is that it allows to prove general theorems in a concise way. For example, Glaßer et al. [10] recently showed that if \( C \) is a class that is balanced-leaf-language definable by a regular language, then all many-one complete problems of \( C \) are polynomial-time many-one autoreducible. This general theorem answered several open questions, since classes like NP, PSPACE, and the levels of the PH are definable in this way.

Moreover, leaf languages allow concise oracle constructions. The background is the BCSV-theorem [5,20] that connects polylog-time reducibility (plt-reducibility) with the robust inclusion of two complexity classes (i.e., the inclusion with respect to all oracles). This connection reduces oracle constructions to their combinatorial core. In particular, neither do we have to care about the detailed stagewise construction of the oracle, nor do we have to describe the particular coding of the single stages.

In this paper we offer a useful generalization of the known leaf-language concepts. Despite of its broader definition, the new concept is convenient and has the nice features we appreciate with traditional leaf languages. It even combines...

* Supported by the Konrad-Adenauer-Stiftung.
certain advantages of single known concepts. We summarize the benefit of the new notion:

1. works with balanced computation trees
2. admits a BCSV-theorem
3. establishes a tight connection between the polynomial-time hierarchy and the Straubing-Thérien hierarchy (the quantifier-alternation hierarchy of the logic \(FO[<\) on words)

The new e-model of leaf languages is inspired by the observation that rejecting paths of nondeterministic computations act as neutral elements. In this sense we allow nondeterministic transducers not only to output single letters, but also to output the empty word \(\varepsilon\) which is the neutral element of \(\Sigma^*\). More precisely, we consider nondeterministic polynomial-time-bounded Turing machines \(M\) such that on every input, every computation path stops and outputs an element from \(\Sigma \cup \{\varepsilon\}\). Let \(M(x)\) denote the computation tree on input \(x\), and define \(\beta_M(x)\) as the concatenation of all outputs of \(M(x)\). For any language \(B\), let \(\text{Leaf}_e^p(B)\) (the e-class of \(B\)) be the class of languages \(L\) such that there exists a nondeterministic polynomial-time-bounded Turing machine \(M\) as above such that for all \(x\),

\[
x \in L \iff \beta_M(x) \in B.
\]

If we demand that \(M\) never outputs \(\varepsilon\), then this defines \(\text{Leaf}_u^p(B)\) (the u-class of \(B\)). If we demand that \(M\) is balanced and never outputs \(\varepsilon\), then this defines \(\text{Leaf}_b^p(B)\) (the b-class of \(B\)). \(M\) is balanced if there exists a polynomial-time computable function that on input \((x, n)\) computes the \(n\)-th path of \(M(x)\).) The notions e-class, u-class, and b-class are extended from a single language \(B\) to a class of languages \(C\) in the standard way: \(\text{Leaf}_e^p(C)\) (the e-class of \(C\)) is the union of all \(\text{Leaf}_e^p(B)\) where \(B \in C\). For a survey on the leaf-language approach we refer to Wagner [21].

It is immediately clear that the u-model and the b-model are restrictions of the e-model.

\[
\text{Leaf}_u^p(B) \subseteq \text{Leaf}_b^p(B) \subseteq \text{Leaf}_e^p(B)
\]

Moreover, it is intuitively clear that the presence of the neutral element \(\varepsilon\) gives the class \(\text{Leaf}_e^p(B)\) some inherent nondeterministic power which makes \(\text{Leaf}_e^p(B)\) seemingly bigger than P. We will discuss this issue and we will identify \(\text{UP} \cap \text{coUP}\) as a lower bound (we obtain stronger bounds if we restrict to regular languages \(B\)). The advantage of the e-model over the u-model is its simplicity: In the e-model we can assume balanced computation trees which in turn leads to easy plt-reductions. The advantage over the b-model is the established tight connection between the polynomial-time hierarchy and the Straubing-Thérien hierarchy, a well-studied hierarchy of regular languages. Glaßer [9] shows that such a connection does not hold for the b-model. This connection within the e-model makes it possible to exactly characterize leaf-language classes in the environment of NP.

In order to describe our results we have to define the levels of the Straubing-Thérien hierarchy (STH). In the scope of this paper it suffices to summarize