On Measuring Non-Recursive Trade-Offs

Hermann Gruber, Markus Holzer, and Martin Kutrib
Institut für Informatik, Universität Giessen,
Arndtstraße 2, 35392 Giessen, Germany
E-Mail: {gruber,holzer,kutrib}@informatik.uni-giessen.de

In computer science in general, and also in the particular field of descriptional complexity, we try to classify problems and mechanisms according to different aspects of their tractability. Often the first distinction we make in such a classification is to check whether a problem admits an effective solution at all. If so, we usually take a closer look and analyze the inherent complexity of the problem. But undecidable problems can also be compared to each other, using the toolkit provided by computability theory. Here, it turns out that most naturally occurring problems are complete at some level of the arithmetic (or analytic) hierarchy. This has been a rather successful approach to understand the nature of many undecidable problems we encounter in various computational settings. As for decision problems, there are conversion problems between different models that cannot be solved effectively. Indeed, they evade solvability *a forteriori* because the size blow-up caused by such a conversion cannot be bounded above by any recursive function. This phenomenon, nowadays known as *non-recursive trade-off*, was first observed by Meyer and Fischer [12] between nondeterministic pushdown automata and finite automata. Previously, it had been known that every deterministic pushdown automaton accepting a regular language can be converted into an equivalent finite automaton of at most triply-exponential size. In contrast, Meyer and Fischer showed that if we replace "deterministic pushdown automaton" with "nondeterministic pushdown automaton," then the maximum size blow-up can no longer be bounded by any recursive function. Since that time there has been a steadily growing list of results where this phenomenon has been observed, e.g., [1, 3, 4, 5, 6, 7, 8, 10, 11, 14, 15, 16]. In [9] a survey is given that also presents a few general proof techniques for proving such results. While it seems to be clear that non-recursive trade-offs usually sprout at the wayside of the crossroads of (un)decidability, in many cases proving such trade-offs apparently requires ingenuity and careful automata constructions. While apparently we cannot get rid of this altogether, here we identify general criteria where non-recursive trade-offs can be directly read off, provided certain basic (un)decidability results about the descriptional systems under consideration are known. The present work aims at making the first steps in paralleling the successful development of the abstract theory of languages, and in building a theory with unified proofs of many non-recursive trade-off results appearing in the literature. We can show the following two results, which allows us to deduce that certain trade-offs between descriptional systems[1] are non-recursive[2] in a very easy way.

Theorem 1 *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two descriptional systems that are effective full trios. If the infiniteness problem for $\mathcal{S}_1$ is not semi-decidable and the infiniteness problem for $\mathcal{S}_2$ is decidable, then the trade-off between $\mathcal{S}_1$ and $\mathcal{S}_2$ is non-recursive.*

Here a descriptional system is called an *effective trio*, if it is effectively closed under λ-free morphism, inverse morphism and intersection with regular languages. If it is also effectively closed under general morphism, we speak of an *effective full trio.*

Theorem 2 *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two descriptional systems that are effective trios. If $\mathcal{S}_1$ has a decidable word problem but an undecidable emptiness problem, and $\mathcal{S}_2$ has a decidable emptiness problem, then the trade-off between $\mathcal{S}_1$ and $\mathcal{S}_2$ is non-recursive.*

Besides new proof techniques in this domain, the present work also aims to provide a finer classification of such non-recursive trade-offs, in a similar vein to what has been done in the classification of undecidable problems. We prove bounds on the trade-off function f that serves as a least upper bound for the increase in complexity when changing from a descriptor in $\mathcal{S}_1$ to an equivalent descriptor in $\mathcal{S}_2$. Here, it turns out that the complexity[3] of the problem of the $\mathcal{S}_2$-ness of $\mathcal{S}_1$ descriptors influences the growth rate of f.

Theorem 3 *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two descriptional systems. The problem of determining for a given descriptor $D_1 \in \mathcal{S}_1$ whether the language $L(D_1)$ belongs to $\mathscr{L}(\mathcal{S}_2)$, i.e., the $\mathcal{S}_2$-ness of $\mathcal{S}_1$*

[1]A *descriptional system* $\mathcal{S}$ is a recursive set of non-empty finite descriptors, such that each descriptor $D \in \mathcal{S}$ describes a formal language $L(D)$, and if $L(D)$ is recursive (recursively enumerable), then there exists an effective procedure to convert D into a Turing machine that decides (semi-decides) $L(D)$. We always assume that a descriptional system is associated with a reasonable size measure. Here a *complexity (size) measure* for $\mathcal{S}$ is a total, recursive function $c : \mathcal{S} \to \mathbb{N}$ such that for any alphabet A, the set of descriptors in $\mathcal{S}$ describing languages over A is recursively enumerable in order of increasing size, and does not contain infinitely many descriptors of the same size.

[2]Let $\mathcal{S}_1$ be a descriptional systems with complexity measure c_1, and $\mathcal{S}_2$ be descriptional systems with complexity measure c_2. A total function $f : \mathbb{N} \to \mathbb{N}$, with $f(n) \geq n$, is said to be an *upper bound* for the increase in complexity when changing from a descriptor in $\mathcal{S}_1$ to an equivalent descriptor in $\mathcal{S}_2$, if for all $D_1 \in \mathcal{S}_1$ with $L(D_1) \in \mathscr{L}(\mathcal{S}_2)$ there exists a $D_2 \in \mathcal{S}_2(L(D_1))$ such that $c_2(D_2) \leq f(c_1(D_1))$. If there is no recursive upper bound, the trade-off is said to be *non-recursive.*

[3]In particular we consider the *arithmetic hierarchy* [13], which is defined as follows:

$$\begin{aligned} \Sigma_1 &= \{ L \mid L \text{ is recursively enumerable} \}, \\ \Sigma_{n+1} &= \{ L \mid L \text{ is recursively enumerable in some } A \in \Sigma_n \}, \end{aligned}$$

for $n \geq 1$. Here, a language L is said to be recursively enumerable in some B if there is a Turing machine with oracle B that semi-decides L. Let Π_n be the complement of Σ_n, i.e., $\Pi_n = \{ L \mid \overline{L} \text{ is in } \Sigma_n \}$. Moreover, let $\Delta_n = \Sigma_n \cap \Pi_n$, for $n \geq 1$. Observe that $\Delta_1 = \Sigma_1 \cap \Pi_1$ is the class of all recursive sets. Completeness and hardness are always meant with respect to many-one reducibilities $\leq_m$, if not otherwise stated. Let K denote the *halting set*, i.e., the set of all encodings of Turing machines that accept their own encoding. For any set A define $A' = K^A$ to be the *jump* or *completion* of A, where K^A is the *A-relativized halting set*, which is the set of all encodings of Turing machines with oracle A that accept their own encoding, and define $A^{(0)} = A$ and $A^{(n+1)} = (A^{(n)})'$, for $n \geq 0$. Furthermore we use $\leq_T$ to refer to Turing reducibility.

descriptors, can be solved in Σ_2*, if* both $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are recursive. In case* at least one *descriptional system is not recursive (but recursively enumerable) the problem can be solved in* Σ_3*.*

This theorem can be utilized to prove an upper bound when changing from one system to another one.

Theorem 4 *Let* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *be two descriptional systems. If both* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are recursive, then there is a total function* $f : \mathbb{N} \to \mathbb{N}$ *that serves as an upper bound for the increase in complexity when changing from a descriptor in* $\mathcal{S}_1$ *to an equivalent descriptor in* $\mathcal{S}_2$*, satisfying* $f \leq_T \emptyset''$*. In case* at least one *descriptional system is not recursive (but recursively enumerable) the function* $f : \mathbb{N} \to \mathbb{N}$ *can be chosen to satisfy* $f \leq_T \emptyset'''$*.*

What about lower bounds on the trade-off function f? In fact, we show that there is a relation between the function f and the equivalence problem between $\mathcal{S}_1$ and $\mathcal{S}_2$ descriptors, in the sense that, whenever the former problem becomes easy, the latter is easy too.

Theorem 5 *Let* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *be two descriptional systems and* $f : \mathbb{N} \to \mathbb{N}$ *a total function that serves as an upper bound for the increase in complexity when changing from a descriptor in* $\mathcal{S}_1$ *to an equivalent descriptor in* $\mathcal{S}_2$*. Then we have: If both descriptional systems are recursive and* $f \leq_T \emptyset'$*, then the* $\mathcal{S}_2$*-ness of* $\mathcal{S}_1$ *descriptors is recursive in* $\emptyset'$*. If at least one descriptional system is not recursive (but recursively enumerable) and* $f \leq_T \emptyset''$*, then the* $\mathcal{S}_2$*-ness of* $\mathcal{S}_1$ *descriptors is recursive in* $\emptyset''$*.*

Thus, we can show that only *two* types of non-recursive trade-offs within the recursively enumerable languages exist! First consider the context-free grammars and the right-linear context-free grammars (or equivalently finite automata) as descriptional systems. Thus, we want to consider the trade-off between context-free languages and regular languages. In [12] it was shown that this trade-off is non-recursive. By Theorem 4, one can choose the upper bound function f such that $f \leq_T \emptyset''$. On the other hand, if $f \leq_T \emptyset'$, then by Theorem 5 we deduce that checking regularity for context-free grammars is recursive in $\emptyset'$ and hence belongs to Δ_2. This is a contradiction, because in [2] this problem is classified to be Σ_2-complete. So, we obtain a non-recursive trade-off somewhere in between $\emptyset''$ and $\emptyset'$, that is, $f \leq_T \emptyset''$ but $f \not\leq_T \emptyset'$.

References

[1] I. Borchardt. Nonrecursive tradeoffs between context-free grammars with different constant ambiguity. Master's thesis, Universität Frankfurt, 1992. (in German).

[2] D. F. Cudia. The degree hierarchy of undecidable problems of formal grammars. *Symposium on Theory of Computing (STOC 1970)*, 10–21, 1970.

[3] J. Goldstine, M. Kappes, Ch. M. R. Kintala, H. Leung, A. Malcher, and D. Wotschke. Descriptional complexity of machines with limited resources. *J. UCS*, 8:193–234, 2002.

[4] J. Hartmanis. On the succinctness of different representations of languages. *SIAM J. Comput.*, 9:114–120, 1980.

[5] J. Hartmanis. On Gödel speed-up and succinctness of language representations. *Theoret. Comput. Sci.*, 26:335–342, 1983.

[6] Ch. Herzog. Pushdown automata with bounded nondeterminism and bounded ambiguity. *Theoret. Comput. Sci.*, 181:141–157, 1997.

[7] C. Kapoutsis. From $k + 1$ to k heads the descriptive trade-off is non-recursive. *Descriptional Complexity of Formal Systems (DCFS 2004)*, 213–224, 2004.

[8] M. Kutrib. On the descriptional power of heads, counters, and pebbles. *Theoret. Comput. Sci.*, 330:311–324, 2005.

[9] M. Kutrib. The phenomenon of non-recursive trade-offs. *Int. J. Found. Comput. Sci.*, 16:957–973, 2005.

[10] A. Malcher. Descriptional complexity of cellular automata and decidability questions. *J. Autom., Lang. Comb.*, 7:549–560, 2002.

[11] A. Malcher. On the descriptional complexity of iterative arrays. *IEICE Trans. Inf. Syst.*, E87-D:721–725, 2004.

[12] A. R. Meyer and M. J. Fischer. Economy of description by automata, grammars, and formal systems. Symposium on Switching and Automata Theory (SWAT 1971), 188–191, 1971.

[13] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.

[14] E. M. Schmidt and Th. G. Szymanski. Succinctness of descriptions of unambiguous context-free languages. *SIAM J. Comput.*, 6:547–553, 1977.

[15] B. Sunckel. On the descriptional complexity of metalinear CD grammar systems. *Descriptional Complexity of Formal Systems (DCFS 2004)*, 260–273, 2004.

[16] L. G. Valiant. A note on the succinctness of descriptions of deterministic languages. *Inform. Control*, 32:139–145, 1976.