# Automata and logic — descriptive set theory

Michał Skrzypczak
*http://www.mimuw.edu.pl/~mskrzypczak/*

Verification and model-checking is my general area of research. I'm intersted in the so-called *formal methods*, which analyse properties of a given system basing on mathematical tools. A fundamental instance of this approach is automata theory, where one considers finite-state systems and specifications definable in monadic second-order (MSO) logic. Classical results of Elgot, Trakhtenbrot, Büchi, Landweber, Rabin, etc. showed decidability of many problems of model-checking, verification, and synthesis in this setting.

In my research I focus on models of *reactive systems* where the computations under consideration are infinite. Thus, an execution of a system is an infinite sequence of states — an infinite word. When one considers all possible executions at the same time, they naturally arrange in the form of an infinite tree. Thanks to the results of Büchi and Rabin, the satisfiability problem for MSO is decidable for both these classes of structures. Additionally, the spaces of all infinite words and infinite trees bear a topological structure making them homeomorphic with the Cantor set. This allows us to mix combinatorial methods of automata theory and methods of classical topology.

The main goal of the subject under discussion is the following meta-question: given a model of reactive machines, understand its expressive power. A more concrete example is: is it possible to transform every non-deterministic machine of the given model into an equivalent deterministic one? Many tools for solving such questions are given by descriptive set theory. This discipline studies how complex a given set is, where the complexity is measured in set theoretic operations needed to define the given set. Thus, we start from some family of *basic* sets and construct more and more complex sets by applying some standard operations (e.g. countable unions and countable intersections). This can be see as a set-theoretic counterpart of the complexity theory. However, the advantage of descriptive set theory is that rather simple arguments show that all the hierarchies under consideration are strict. Thus, we don't need to rely on assumptions like PTime $\neq$NP.

A generic example of a reasoning that involves descriptive set theory is the following. We want to know is it possible to determinise every machine in a class $\mathcal{M}$ into an equivalent deterministic one. We define a particular example of a machine $M_0 \in \mathcal{M}$ and prove that the set of infinite words defined by $M_0$ (denoted $\mathrm{L}(M_0)$) is hard from the point of view of descriptive set theory. Additionally, we prove that if $M' \in \mathcal{M}$ is deterministic, then there exists an easy description of the set $\mathrm{L}(M')$. Therefore, the set $\mathrm{L}(M_0)$ cannot be recognised by any deterministic machine $M' \in \mathcal{M}$ by the strictness of the respective hierarchy. Thus, we obtained a purely automata-theoretic result on determinisability using tools of descriptive set theory. Some examples of results of the above kind that I have took part in can be found in [Skr16].

## References

[Skr16] Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory - Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016.