Language Theory and Infinite Graphs

Colin Stirling
University of Edinburgh, UK

Automata and language theory study finitely presented mechanisms for generating languages. A language is a family of words. The Chomsky hierarchy of languages can be generated using grammars or using automata. At the lowest level are regular languages which are also generated by finite-state automata. At the next level are context-free languages. These are generated by context-free grammars and also by pushdown automata. Beyond this are the context-sensitive languages and the recursively enumerable languages, generated by linear bounded Turing machines and Turing machines.

A slight shift in focus is very revealing. Instead of grammars and automata as language generators, one views them as propagators of possibly infinite labelled transition graphs. This is our starting point. We shall examine various kinds of infinite graph, concentrating on pushdown automata and context-free grammars and also consider bisimulation equivalence as an alternative to language equivalence.

The main goal of the lectures is to provide decision procedures for infinite state graphs. We concentrate on proving decidability of language equivalence using both graph theoretic and combinatorial arguments. The main result to be examined is decidability of the DPDA equivalence problem: that language equivalence is decidable for deterministic context-free languages. Despite intensive work throughout the late 1960s and 1970s, this problem remained unsolved until 1997 when Sénizergues announced a positive solution. His proof consisted of two semi-decision procedures, and so there was no complexity bound on the procedure.

In the lectures we provide a deterministic decision procedure with a primitive recursive upper bound. We shall also briefly discuss the open problem whether language equivalence is decidable for deterministic higher-order grammars. For background reading on the definitions of language equivalence, context free grammars and (deterministic) pushdown automata see, for instance, [1].

References

1. J. Hopcroft, J. Ullman. *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979