Mathematical Structures in Formal Methods, *MSFM*
Handout for Lecture 1 (2018/4/26)

Ichiro Hasuo, NII/SOKENDAI, Tokyo, Japan `http://group-mmm.org/∼ichiro`

1 Course Information

1.1 Webpage

Find latest notices at `http://group-mmm.org/~ichiro/COURSE_msfm2018.html`.

1.2 General Information

Thursday 10.45–12.15
Room: 1212, NII 12th Floor
Lectures are in English
Evaluation: Term-end report 80%, other reports 20% (subject to change)

1.3 Overview and Objectives

Formal methods refer to a body of mathematical techniques used for guaranteeing correctness of computer systems. This course introduces the mathematical foundation of formal methods. Our technical focus will be on automata theory, especially on automata that classify infinite words.

- In-depth understanding of basic automata theory
- Understanding of use of automata theory for formal methods purposes like verification and synthesis
- Hands-on knowledge on implementation of automata-based algorithms

1.4 Course Structure

(1–3) Automata on finite words, from an algorithmic point of view
(4–6) Automata on infinite words
(7) Linear temporal logic (LTL) as a specification language
(8) LTL model checking via translation to automata
(9–10) Parity games and their decision procedure
(11–12) Model checking and synthesis by parity games
(13–15) Probabilistic systems and probabilistic LTL model checking

1.5 Teaching Methods

Important: it is *outside the lecture room* that you will actually learn things. The lectures merely provide guidance and support. Spend time with the references.

Topics for the next lecture will be announced in handouts, in advance. It is highly recommended that you do some preparatory reading before lectures. (A quick glance is better than none.)

Report assignments will be given occasionally; they consist of review questions (for what has been already covered in lectures) and preparatory questions (for what is to come).

1.6 Prerequisites

Familiarity with the following topics is desirable (but not mandatory): propositional logic, computational complexity, formal language theory

1.7 References

- Moshe Y. Vardi: An Automata-Theoretic Approach to Linear Temporal Logic. Banff Higher Order Workshop 1995: 238-266 (Used in the lecture)
- S. Demri and P. Gastin. Specification and Verification using Temporal Logics. In Modern applications of automata theory, IISc Research Monographs 2, chapter 15, pages 457-494. World Scientific, 2012. (Used in the lecture, for temporal logic syntax and semantics)
- Jurdzinski M. (2000) Small Progress Measures for Solving Parity Games. In: Reichel H., Tison S. (eds) STACS 2000. STACS 2000. Lecture Notes in Computer Science, vol 1770. Springer, Berlin, Heidelberg (Used in the lecture, for parity games)
- Wilke, Thomas. Alternating tree automata, parity games, and modal μ-calculus. Bull. Belg. Math. Soc. Simon Stevin 8 (2001), no. 2, 359–391. (Comprehensive reference for the last part of the course)
- Christel Baier and Joost-Pieter Katoen. 2008. Principles of Model Checking (Representation and Mind Series). The MIT Press. (A thick textbook. Chapter 10 will be used for probabilistic model checking)

1.8 Schedule

2018.4.19 (Thu) No lecture (due to the lecturer's absence)
2018.4.26 (Thu)
2018.5.10 (Thu)
2018.5.17 (Thu)
2018.5.24 (Thu)
2018.5.31 (Thu)
2018.6.7 (Thu)
2018.6.14 (Thu)
2018.6.21 (Thu)
2018.6.28 (Thu)
2018.7.5 (Thu)
2018.7.12 (Thu)
2018.7.19 (Thu)
2018.7.26 (Thu)
2018.8.2 (Thu)

Some will be canceled due to the lecture's trips.

2 Today's Lecture

Introduction to the course, introduction to automata theory, especially the material *not* covered in [Vardi, 1995]. Topics: pumping lemma (in detail), powerset construction (in detail), regular expression and the Kleene theorem (sketch), the Myhill–Nerode theorem and minimization (sketch).

3 Next Lecture

[Vardi, 1995, Section 2.1]

4 Report Assignments

Due: the beginning of the next lecture. Hand in a hard copy, or submit electronically to `i.hasuo@acm.org`.

1. Prove that the following language is not regular: $L = \{ww^R \mid w \in \{0,1\}^*\}$, where $(a_1a_2\dots a_n)^R := a_na_{n-1}\dots a_1$.
2. Sketch an algorithm for the following task. Input: an NFA M. Output: if $L(M) = \emptyset$ or not.